CSDL-T-912

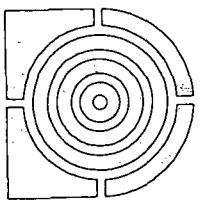
A LATERAL GUIDANCE ALGORITHM TO REDUCE THE POST-AEROBRAKING BURN REQUIREMENTS FOR A LIFT-MODULATED ORBITAL TRANSFER VEHICLE

by

Glen Charles Herman

June 1986

M.I.T. Masters Thesis



D3588421

The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

A LATERAL GUIDANCE ALGORITHM TO REDUCE THE POST-AEROBRAKING BURN REQUIREMENTS FOR A LIFT-MODULATED ORBITAL TRANSFER VEHICLE

by

GLEN CHARLES HERMAN

B.S., University of Michigan
(1984)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN
AERONAUTICS AND ASTRONAUTICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 1986

Signature of Author Glen Charles Herman
Department of Aeronautics and Astronautics
May 16, 1986

Certified by Richard H. Battin
Dr. Richard H. Battin
Adjunct Professor of Aeronautics and Astronautics, M.I.T.
Thesis Supervisor

Certified by Timothy J. Brand
Timothy J. Brand
Technical Supervisor, CSDL

Accepted by _____
Professor Harold Y. Wachman
Chairman, Departmental Graduate Committee

A LATERAL GUIDANCE ALGORITHM TO REDUCE THE POST-AEROBRAKING BURN REQUIREMENTS FOR A LIFT-MODULATED ORBITAL TRANSFER VEHICLE

by

Glen Charles Herman

Submitted to the Department of Aeronautics and Astronautics
on May 16, 1986, in partial fulfillment of the
requirements for the Degree of Master of Science
in Aeronautics and Astronautics

ABSTRACT

A lateral guidance algorithm which controls the location of the line of intersection between the actual and desired orbital planes (the hinge line) is developed for the aerobraking phase of a lift-modulated orbital transfer vehicle. The on-board targeting algorithm associated with this lateral guidance algorithm is simple and concise which is very desireable since computation time and space are limited on an on-board flight computer. A variational equation which describes the movement of the hinge line is derived. Simple relationships between the plane error, the desired hinge line position, the position out-of-plane error, and the velocity out-of-plane error are found. A computer simulation is developed to test the lateral guidance algorithm for a variety of operating conditions. The algorithm does reduce the total burn magnitude needed to achieve the desired orbit by allowing the plane correction and perigee-raising burn to be combined in a single manuever. The algorithm performs well under vacuum perigee dispersions, pot-hole density disturbances, and thick atmospheres. The results for many different operating conditions are presented.

Thesis Supervisor: Dr. Richard H. Battin

Title: Adjunct Professor of Aeronautics and Astronautics

Technical Supervisor: Timothy J. Brand

Title: Division Leader, Charles Stark Draper Laboratory, Inc.

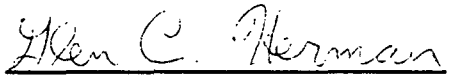
ACKNOWLEDGEMENT

I would like to thank my parents for their continuous support which helped me reach this point in my educational career. Next, I wish to convey my thanks to John Higgens and Tim Brand of The Charles Stark Draper Laboratory who provided me with valuable technical support and guidance. A special thanks to my friends Steve, Mark, Rene, and Sam for providing help in preparing this thesis. My deepest appreciation to Florie for making this thesis look as great as it does. Lastly, I would like to thank my thesis supervisor, Dr. Battin, for his patience and understanding.

This research was funded by The Charles Stark Draper Laboratory, Inc. under contract number NAS9-17560.

Publication of this report does not constitute approval by the Charles Stark Draper Laboratory, Inc. of the findings and conclusions contained herein. It is published for the exchange and simulation of ideas.

I hereby assign my copyright of this thesis to The Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts.



Glen C. Herman

Permission is hereby granted to The Charles Stark Draper Laboratory, Inc. and to the Massachusetts Institute of Technology to reproduce and to publicly distribute copies of this thesis in whole or in part.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

Orbital transfer vehicles (OTV's) have been the focus of considerable research efforts in recent years. The main mission of the OTV is to carry payloads between low Earth orbit (LEO) and geosynchronous orbit (GEO). The velocity decrement necessary in transferring from GEO to LEO can either be done all propulsively or assisted by using aerobraking. Aerobraking is a maneuver in which the OTV enters the Earth's upper atmosphere and uses the aerodynamic forces generated to reduce its velocity and control its trajectory before returning to LEO. Even though the OTV still needs to use propulsive maneuvers to attain the desired circular orbit when using aerobraking, a major portion of the necessary velocity decrement (roughly 8000 ft/sec) is attained with no expenditure of fuel if the OTV uses aerobraking. This fuel savings creates more payload space on the OTV and increases the payload weight which the OTV can carry. This increase in payload capacity and the

reduction in fuel requirements makes an OTV which uses aerobraking more desirable than one which just uses propulsive maneuvers.

Drag modulation and lift modulation are the two basic approaches in designing an aerobraking OTV. A drag-modulated OTV only uses drag to control its trajectory and requires that the lift forces generated are small. The drag is modulated by changing the OTV drag coefficient (C_D) and cross sectional area (A) by inflating and deflating a balloon-like bag, called a ballute, attached to the vehicle [1,2]. The advantage of this approach is that the vehicle's structural design can be more symmetric and no attitude control is needed; however, the actual control of the ballute's shape in the upper atmosphere is not a trivial problem. Unfortunately, drag modulation does not provide a way to adjust the orbital plane since the components of the velocity and position vectors normal to the desired orbital plane (ie. out-of-plane errors) can not be controlled.

The OTV considered in this thesis flies with a constant angle of attack and a near constant L/D and is aerodynamically similar to the Apollo command module. The OTV trajectory is controlled by modulating the lift direction with roll adjustments to regulate the lift component in the current orbital plane (ie. in-plane). The roll angle is varied by using the OTV roll jets while in the atmosphere. The presence of lift forces not only allows control of the trajectory, depth of pene-

tration into the atmosphere, and hence the velocity decrement, but also enables the vehicle to control its velocity and position out-of-plane errors. Therefore, the lift-modulated OTV can adjust its orbital plane unlike the drag-modulated OTV. This is a very important capability since the velocity increment needed to correct just a one degree plane error for a 150 nautical mile circular orbit is 443 ft/sec.

There are many different guidance algorithms for controlling a lift-modulated OTV during the aerobraking maneuver [3,4,5]. In general these algorithms solve for the required lift component in the current orbital plane (ie. in-plane) and the roll angle needed to achieve it. Therefore, there will be some lift component normal to the current orbital plane (ie. out-of-plane) remaining which will change the orbital plane of the OTV generating a plane error. Another common characteristic of these algorithms is that only the magnitude of the roll angle is specified and not its sign. This extra degree of freedom can be exploited by developing an appropriate lateral guidance algorithm. The purpose of this lateral guidance algorithm would be to minimize the velocity increments normal to the current orbital plane needed to place the OTV in its target orbit. Particularly, the large velocity increment needed to correct plane errors can be greatly reduced by using a lateral guidance algorithm which controls the orientation of the OTV orbital plane.

1.2 Motivation

The OTV considered in this thesis controls its trajectory by regulating the in-plane lift component. The magnitude of the in-plane lift component is adjusted by directing a portion of the lift vector out-of-plane. Plane errors are inevitable when out-of-plane lift forces are present, because the out-of-plane lift forces will change the orbital plane of the OTV. Since plane errors can only be corrected when the OTV is at the line of intersection between the desired and actual orbital planes, the plane error can not be nulled during the aerobraking maneuver. The plane error must be corrected impulsively at a high cost when the OTV leaves the atmosphere and is at the line of intersection between the desired and actual orbital planes. Therefore, designing a lateral guidance algorithm which reduces the velocity increment needed to correct the plane error is desireable.

There are several different approaches in designing a lateral guidance algorithm which will reduce the velocity increments needed to correct plane errors. One obvious approach is to develop an algorithm which controls the size of the plane error. The plane error consists of errors in both the inclination and ascending node. A lateral guidance algorithm which controls the plane error is described in reference [6]; this algorithm uses roll reversals to minimize the plane error by attempting to zero the velocity out-of-plane error (ie. the velocity

component normal to the desired plane). The plane error, however, can not realistically be zeroed, since the number of roll reversals allowed is limited. This limitation is due to roll jet fuel consumption and structural dynamics considerations. Also, limiting the number of roll reversals performed is desireable, since roll reversals require the in-plane lift component to differ from the commanded in-plane value temporarily which might generate undesireable transients.

Further insight in developing a lateral guidance algorithm can be gained by examining the orbital mechanics of the post-aerobraking maneuvers. The OTV performs a deorbit burn at GEO which puts it in an elliptical transfer orbit with an apogee altitude of 19,323 nautical miles and a vacuum perigee altitude of 41 nautical miles. The aerobraking guidance law is designed to reduce the apogee altitude to 150 nautical miles by the time the OTV exits the atmosphere. However, atmospheric density disturbances, guidance errors, and navigation errors will make the actual apogee altitude slightly different from the desired value of 150 nautical miles. Once the OTV leaves the atmosphere, its velocity must be adjusted in order to attain the desired target orbit. The required changes in velocity are referred to as burns. The post-aerobraking maneuvers consist of three separate burns. The perigee-raising burn is made at apogee and raises the perigee to the desired circular altitude. The circularization trim burn is made at perigee and circularizes the orbit. The circularization trim burn is needed to correct

for the difference between the actual post-aerobraking apogee altitude and the desired value. The plane correction trim burn is performed at the line of intersection between the target and actual orbital planes, known as the hinge line, and corrects the plane error. The total burn magnitude is the sum of these three burns. The perigee-raising burn accounts for a large majority of the total burn magnitude while the two trim burns only make up a small fraction of the total burn magnitude.

The velocity increment needed to correct the plane change can be reduced further by correcting some of the plane error with the perigee-raising burn. A plane change can be made with a very small increase in the burn magnitude simply by placing a portion of the perigee-raising burn vector out-of-plane (ie. normal to the current orbital plane). Combining a plane change burn with another type of burn is called a dog-leg maneuver. A dog-leg maneuver places the perigee-raising burn vector out-of-plane and makes an in-plane and out-of-plane orbital correction with just one burn. By performing a dog-leg maneuver, small plane errors can be corrected at little cost. A maximum velocity increment saving of 39.2 ft/sec can be achieved by using a dog-leg maneuver to correct a 0.1 degree plane error when circularizing at apogee from an initial elliptical orbit with an apogee altitude of 150 nautical miles and a perigee altitude of 45 nautical miles. Unfortunately, this maximum saving is only possible when the hinge line and the apsidal line (the line connecting perigee and apogee) coincide. When the apsidal

line is 90 degrees away from the hinge line, a dog-leg maneuver is not possible and the plane error must be corrected entirely with the trim burn.

The on-board targeting algorithm which finds the minimum total burn magnitude is extremely complicated when only the plane error magnitude is controlled by the lateral guidance algorithm. Only part of the plane error can be corrected by a dog-leg maneuver since the apsidal line and the hinge line will not necessarily coincide. The amount of plane error to be corrected by the dog-leg maneuver depends on the angle between the hinge line and apsidal line. The proportion of the plane error corrected by a dog-leg maneuver increases as the angle between the hinge line and apsidal line decreases. An analytical method for determining the portion of the plane error to correct on the dog-leg maneuver which will minimize the sum of the burn magnitudes is extremely difficult to develop if it exists at all. Therefore, an iteration process is used to determine the plane change made by the dog-leg maneuver which produces the minimum total burn magnitude. This iteration process is computationally slow and not very efficient in the on-board flight computer.

Significant fuel savings can be attained if the burn magnitude required to correct plane errors can be reduced. The goal of this thesis is to develop a lateral guidance algorithm for aerobraking which controls the size of the plane error and then the location of the hinge

line. This approach in designing a lateral guidance algorithm is advantageous, because it not only reduces the required velocity increments but also simplifies the on-board targeting algorithm. If the hinge line and the apsidal line are assumed to coincide in designing the on-board targeting algorithm, a complex iteration process is no longer needed and the plane error can be corrected completely with the dog-leg maneuver. Any residual plane errors which occur due to errors in this assumption can be corrected with a small trim burn. This simple and concise burn sequence algorithm is desirable since it will be computationally fast in the OTV on-board flight computer.

1.3 Thesis Outline

Chapter 2 provides background information necessary to develop the lateral guidance algorithm. Equations are derived which explain the behavior of the hinge line. Relationships are found between the plane error, the velocity and position out-of-plane errors, and the location of the hinge line. The calculation and selection of the guidance control parameters are discussed. The aerobraking guidance law used when testing this lateral guidance algorithm is presented. Finally, the post-aerobraking burn sequence algorithm is explained.

Chapter 3 contains the complete description of the lateral guidance algorithm development. An overview of the different segments of the algorithm is given. Then the development of each segment is examined in detail.

Chapter 4 analyzes and presents the test results from the computer simulations performed on the algorithm. The computer programs used in the simulation are presented. The reference trajectories flown by the OTV are described. The testing methodology and the performance criteria are discussed. Finally, the results of the numerous tests are given.

Chapter 5 summarizes the conclusions drawn from this thesis and recommends areas of continued research to improve this lateral guidance algorithm.

Appendix A contains the computer source code for the lateral guidance algorithm and the other programs used in the simulation.

CHAPTER 2

FUNDAMENTALS OF THE LATERAL GUIDANCE LOGIC

2.1 Derivation Of The Hinge Line Rate Equation

A fundamental understanding of the variational behavior of the hinge line is required to design an efficient lateral guidance algorithm. By knowing the physical processes involved, the lateral guidance control parameters can be easily selected. The location of the hinge line is a function of the orbital elements. Under normal circumstances, when the vehicle is a point mass operating only under the gravitational influence of a spherical body in a two-body system, the orbital elements and the location of the hinge line are constant. However, if the vehicle is subjected to disturbing accelerations, the orbital elements and the location of the hinge line will no longer be constant. Disturbing accelerations are caused by the non-spherical shape of the Earth, the gravitational forces of other bodies outside the two-body system, aerodynamic forces, and other non-gravitational forces. An equation which describes the behavior of the hinge line when the OTV experiences dis-

turbing accelerations is derived using orbital mechanics and the variation of parameters techniques developed in reference [7].

Figure 2.1 shows the coordinate systems and the associated Euler angles used to describe the location of the hinge line. Three rectangular coordinate systems are used to facilitate the derivation of the hinge line variational equation. The hinge line coordinate system is defined by unit vectors in the direction of the hinge line (i_n) and the angular momentum vector (i_h) of the current orbit, while the direction of the third unit vector (i_m) is chosen to complete the right-handed coordinate system. The apsidal line coordinate system is defined by unit vectors along the apsidal line (i_e) and the angular momentum vector (i_h) of the actual orbit, while the third unit vector (i_p) is chosen to complete the right-handed coordinate system. Both the apsidal line and hinge line coordinate systems are allowed to rotate relative to inertial space, and, therefore, the direction of their unit vectors can vary with time. Finally, a reference coordinate system which is fixed in inertial space is defined by three unit vectors associated with the reference plane. The first unit vector (i_x) lies along the line of intersection between the reference plane and the equatorial plane and points towards the ascending node. The second unit vector (i_z) is perpendicular to the reference orbit and is positive in the north direction. The third unit vector (i_y) completes the right-handed coordinate system and is in the plane of the reference orbit.

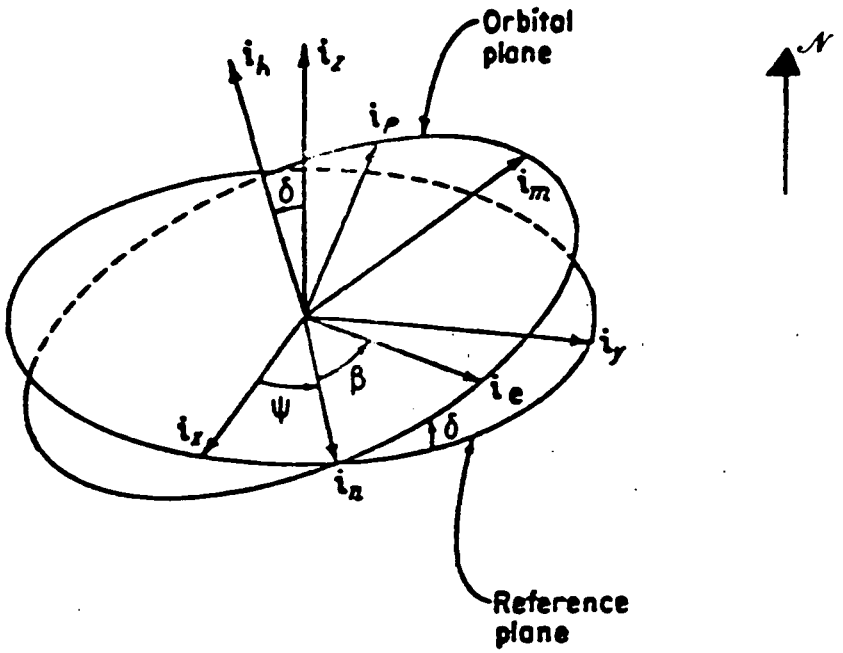


Figure 2.1

Reference Geometry For Hinge Line Coordinate System

The equations which relate the unit vectors of the different coordinate systems are given in reference [7] as:

$$\underline{i}_n = \cos\psi\, \underline{i}_x + \sin\psi\, \underline{i}_y \tag{2.1}$$

$$\underline{i}_m = -\sin\psi\cos\delta\, \underline{i}_x + \cos\delta\, \underline{i}_y + \sin\delta\, \underline{i}_z \tag{2.2}$$

$$\underline{i}_h = \sin\psi\sin\delta\, \underline{i}_x - \cos\psi\sin\delta\, \underline{i}_y + \cos\delta\, \underline{i}_z \tag{2.3}$$

where the plane error (δ) is the angle between the reference and actual orbital planes, β is the angle between the hinge line and the apsidal line, and the longitude of the hinge line (ψ) is the angle between the hinge line and the reference direction $\underline{i}_x$. The angle β is defined to be positive if the apsidal line is south of the reference plane and negative if the apsidal line is north of the reference plane. These three angles δ, β, and ψ are the Euler angles and may be considered as orbital elements.

A variational equation for the longitude of the hinge line (ψ) will describe how the location of the hinge line varies in response to disturbing accelerations. From reference [7], the following rule for deriving variational equations for orbital elements is given:

> Apply the usual rules of differentiation to any two-body indentity. Treat the radius vector ($\underline{r}$) as a constant, the orbital elements as variables, and replace the time rate of change of the velocity vector ($\underline{v}$) by the disturbing acceleration vector ($\underline{a}_d$). [7]

The angular momentum vector ($\underline{h}$) is a good indication of the location of the hinge line since it is always perpendicular to the orbital plane. A variational equation for the longitude of the hinge line will be developed from the variational equation of the angular momentum vector ($\underline{h}$). In inertial reference coordinates, the current angular momentum vector is expressed as:

$$\underline{h} = h\,[\sin\Psi \sin\delta\, \underline{i}_x - \cos\Psi \sin\delta\, \underline{i}_y + \cos\delta\, \underline{i}_z] \qquad (2.4)$$

where h is the magnitude of the angular momentum vector. Applying the differentiation rule stated above to equation 2.4, one obtains:

$$\begin{aligned} d\underline{h}/dt = {} & (\sin\Psi \sin\delta\, \underline{i}_x - \cos\Psi \sin\delta\, \underline{i}_y + \cos\delta\, \underline{i}_z)\, dh/dt \\ & + (\cos\Psi\, \underline{i}_x + \sin\Psi\, \underline{i}_y)\, \sin\delta\, d\Psi/dt \\ & + (\sin\Psi \cos\delta\, \underline{i}_x - \cos\Psi \cos\delta\, \underline{i}_y - \sin\delta\, \underline{i}_z)\, d\delta/dt \end{aligned} \qquad (2.5)$$

or, by substituting equations 2.1, 2.2, and 2.3 into equation 2.5:

$$d\underline{h}/dt = \sin\delta\, d\Psi/dt\, \underline{i}_n - d\delta/dt\, \underline{i}_m + dh/dt\, \underline{i}_h \qquad (2.6)$$

An equation for the variation of Ψ is found by taking the scalar product of equation 2.6 with $\underline{i}_n$ and by rearranging terms:

$$d\Psi/dt = [1/h \sin\delta]\, d\underline{h}/dt \cdot \underline{i}_n \qquad (2.7)$$

In order to replace the scalar product in equation 2.7 with a more convenient and meaningful term, another variational equation must be

derived for the angular momentum vector. The definition of the angular momentum vector is:

$$\underline{h} = \underline{r} \times \underline{v} \tag{2.8}$$

where $\underline{r}$ is the radius vector and $\underline{v}$ is the velocity vector. Applying the differentiation rule to equation 2.8, one obtains:

$$d\underline{h}/dt = \underline{r} \times \underline{a}_d \tag{2.9}$$

where $\underline{a}_d$ is the disturbing acceleration vector. Substituting equation 2.9 into 2.7,

$$d\Psi/dt = (\underline{r} \times \underline{a}_d \cdot \underline{i}_n)/(h \sin \delta) \tag{2.10}$$

but,

$$\underline{r} \times \underline{a}_d \cdot \underline{i}_n = \underline{i}_n \times \underline{r} \cdot \underline{a}_d \tag{2.11}$$

and,

$$\underline{i}_n \times \underline{r}_d = r \sin \eta \, \underline{i}_h \tag{2.12}$$

where η is the angle from the hinge line to the current radius vector. Replacing η with a term involving β is desirable since the ultimate goal of the guidance algorithm is to drive β to zero (ie. make the hinge line and the apsidal line coincide). As seen in Figure 2.2, a simple relationship between β and η is:

$$\eta = \nu - \beta \tag{2.13}$$

where the true anomaly (ν) is the angle between the apsidal line ($\underline{i}_e$) and the radius vector ($\underline{r}$). The minus sign in equation 2.13 is due to the sign convention for β. Substituting equations 2.11, 2.12, and 2.13 into equation 2.7:

$$d\Psi/dt = (1/h \sin \delta) \, [r \sin (\nu - \beta)] \, \underline{i}_h \cdot \underline{a}_d \tag{2.14}$$

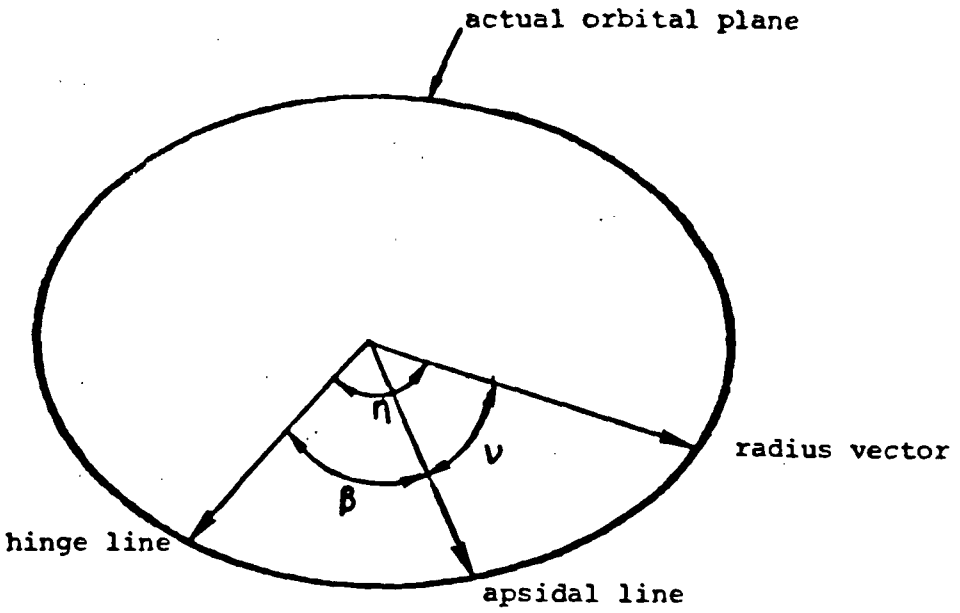


Figure 2.2

Reference Geometry For The Definition of Eta

The scalar product in equation 2.14 represents the component of the disturbing acceleration vector normal to the current orbital plane. For the aerobraking maneuver:

$$\underline{i}_h \cdot \underline{a}_d = \text{LIFTM} \sin \Phi \qquad (2.15)$$

where LIFTM is the magnitude of the lift acceleration and Φ is the roll angle which measures the rotation of the lift vector about the relative velocity vector. The lift vector is straight up and in the current orbital plane when the roll angle is 0.0 degrees and is normal to the current orbital plane when the roll angle is 90 degrees. By substituting equation 2.15 into equation 2.14, one obtains:

$$d\Psi/dt = [r \sin (\nu - \beta)] \text{ LIFTM} \sin \Phi/(h \sin \delta) \qquad (2.16)$$

Equation 2.16 shows the physical forces and variables which effect the movement of the hinge line when the OTV is experiencing aerodynamic forces. Only the lift forces normal to the current orbital plane (ie. out-of-plane) can cause the hinge line to vary position, and the location of the hinge line can be controlled just by changing the sign of the roll angle. Therefore, a lateral guidance algorithm can be designed to control the location of the hinge line, but with the constraint that β does not equal the true anomoly (ie. the hinge line and

the radius vector do not coincide). If β should equal the true anomoly, then the right-hand side of equation 2.16 would equal zero and the location of the hinge line could no longer be changed.

2.2 Plane Error Relations

Parameters which describe the velocity and position out-of-plane errors (ie. the components of the velocity and position vectors normal to the desired orbital plane) are developed in reference [6] as:

$$VY = \underline{i}_z \cdot \underline{v} \tag{2.17}$$

$$RY = \underline{i}_z \cdot \underline{r} \tag{2.18}$$

where RY and VY are the components of the radius and velocity vectors normal to the desired orbital plane respectively. RY and VY depend on the current position and velocity which make them poor indicators of the degree of the out-of-plane errors.

More meaningful indicators of the out-of-plane errors are obtained in reference [6] by defining the following angles:

$$\Theta_R = RY/R \tag{2.19}$$

$$\Theta_V = VY/V \tag{2.20}$$

where R is the magnitude of the radius vector and V is the magnitude of the velocity vector. Θ_R and Θ_V represent the angle between the desired

orbital plane and the radius vector and that plane and the velocity vector, respectively.

As discussed previously, the goal of the lateral guidance algorithm is to make the hinge line and apsidal line coincide (drive β to zero). The right spherical triangle which results from this orbital geometry is shown in Figure 2.3 as viewed from the side. The curent OTV angular position from the hinge line is now the true anomaly (ν), since the hinge line goes through perigee. One side of the spherical triangle represents the actual orbital plane and its length is the value of the true anomaly. The other side of the spherical triangle represents the desired orbital plane, and the angle between this side and the side representing the actual orbital plane is the plane error (δ). Finally, the third side is a great circle which is perpendicular to the desired orbital plane and connects that plane to the current OTV position. The length of this side is Θ_R and the angle it forms with the side representing the actual orbital plane is related to Θ_V as shown in Figure 2.3. The dashed line in Figure 2.3 represents a great circle which passes through the current OTV position and is parallel to the desired orbital plane. The angle between this great circle and the actual orbital plane is Θ_V since the velocity vector is tangent to the actual orbit.

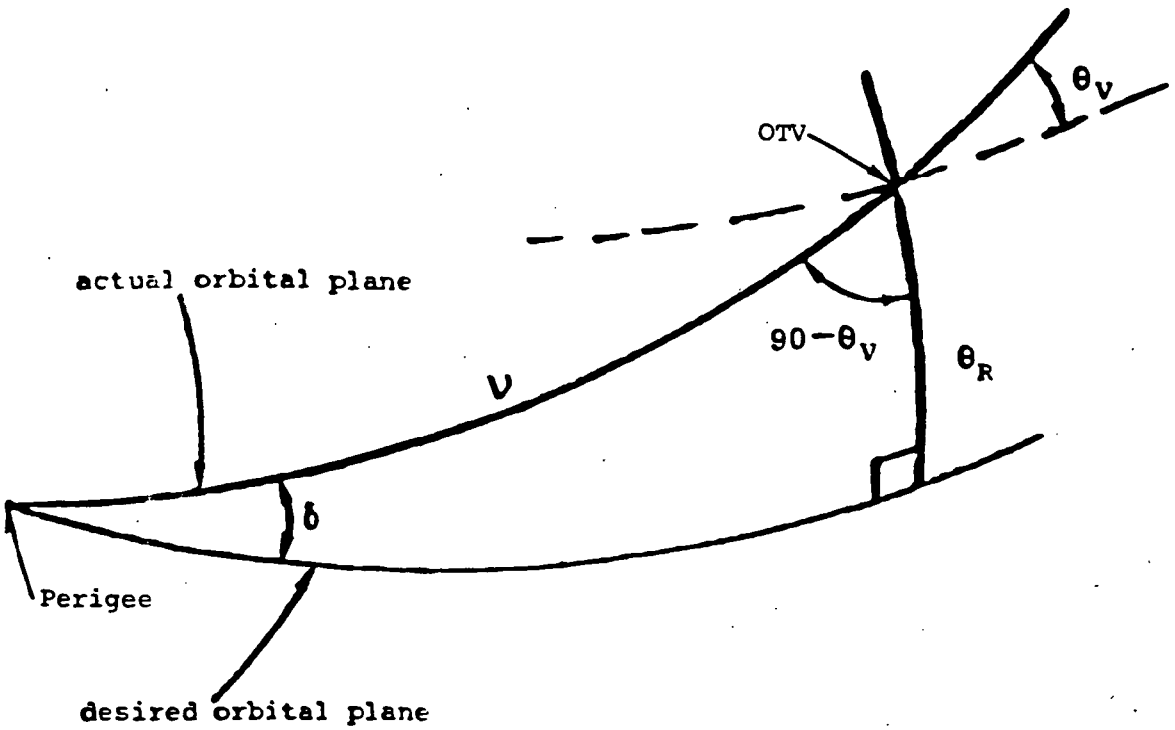


Figure 2.3

Desired Orbital Geometry When The Hinge Line
And Apsidal Line Coincide

A relationship between the plane error and Θ_V and Θ_R is obtained by using the spherical trigonometric relations developed in reference [8]:

$$\cos(\delta) = \sin(90^\circ - \Theta_V)\ \cos\Theta_R \qquad (2.21)$$

but,

$$\sin(90^\circ - \Theta_V) = \cos\Theta_V \qquad (2.23)$$

solving for Θ_V,

$$\Theta_V = \arccos\ [\cos\delta/\cos\Theta_R] \qquad (2.24)$$

Equation 2.4 shows that when Θ_R is very small, the plane error is approximately equal to Θ_V. In the limit as Θ_R approaches zero, the plane error is equal to Θ_V. The relation between the plane error, Θ_V, and Θ_R expressed in equation 2.24 does not depend on the apsidal line and the hinge line coinciding (β being zero).

In reference [6], the desired value of Θ_V was zero in order to minimize the plane error. However, the desired value for Θ_V is different from zero for β to equal zero. Figure 2.3 shows that for a given Θ_R, the desired plane error is given as:

$$\delta_{desired} = \arcsin\ [\sin\Theta_R/\sin\nu] \qquad (2.25)$$

The desired value of Θ_V is now:

$$\Theta_{V\ desired} = \arccos\ [\cos\delta_{desired}/\cos\Theta_R] \qquad (2.26)$$

In the vicinity of perigee, equations 2.25 and 2.26 are ill-defined; therefore, a lateral guidance algorithm based on these equations can

only be used if the vehicle is not near perigee. As in reference [6], a lateral control logic based on a phase plane design could be developed. When the magnitude of Θ_V differs from $\Theta_{V\ desired}$ by a fixed limit, a roll reversal would be commanded. The determination of this fixed limit presents a major problem, because there is no exact relationship between β and the difference between Θ_V and $\Theta_{V\ desired}$.

2.3 Selection And Calculation Of The Control Parameters

A lateral guidance algorithm which causes the hinge line and apsidal line to coincide can either be based on the angle between the hinge line and apsidal line (β) or on θ_v. The magnitude of β is an important control parameter, since it determines the performance of the on-board targeting algorithm (see section 2.5). The angle β will therefore, be the basis for the lateral guidance algorithm developed in this thesis as opposed to θ_v, which was the basis of the algorithm developed in reference [6]. This approach to designing an algorithm is desirable since β can not be directly adjusted when controlling θ_v.

The position of the apsidal line as well as the position of the hinge line varies during the aerobraking maneuver. Unfortunately, the variation of the apsidal line makes the magnitude of β an ambiguous indicator of how the hinge line is moving with respect to the apsidal line. When the rate of change of the apsidal line is greater than the rate of change of the hinge line, the magnitude of β will be increasing even though the hinge line is moving towards the apsidal line. This situation could cause an undesireable roll reversal command since the lateral guidance algorithm is unaware of the direction the hinge line is moving. To avoid this situation, another control parameter is needed which relates the position of the hinge line to some other reference direction.

The argument of perigee (ω), which is an orbital element, represents the angle between the apsidal line and the intersection between the actual orbital plane and the equatorial plane (the line of nodes). A similar parameter for the hinge line is obtained by defining α to be the angle between the hinge line and the line of nodes. Furthermore, the value of α is restricted to lie between 90 degrees and -90 degrees. The angle α is used by the lateral guidance algorithm to measure how the hinge line is moving with respect to the apsidal line. A simple relationship between α, β, and the argument of perigee is:

$$\beta = \alpha - \omega \tag{2.27}$$

as shown in Figure 2.4.

Another important control parameter for the lateral guidance algorithm is the true anomaly (ν) which is also an orbital element. The ability to control the position of the hinge line is severely limited if β and the true anomaly are approximately equal, as discussed in section 2.1. Consequently, one goal of the lateral guidance algorithm is to prevent the hinge line from entering inside a certain region around the current OTV position.

The four control parameters (α, β, the true anomaly, and the argument of perigee) needed by the lateral guidance algorithm can be easily obtained when measurements of the position vector and the velocity vector are available. All the orbital elements can be determined from the

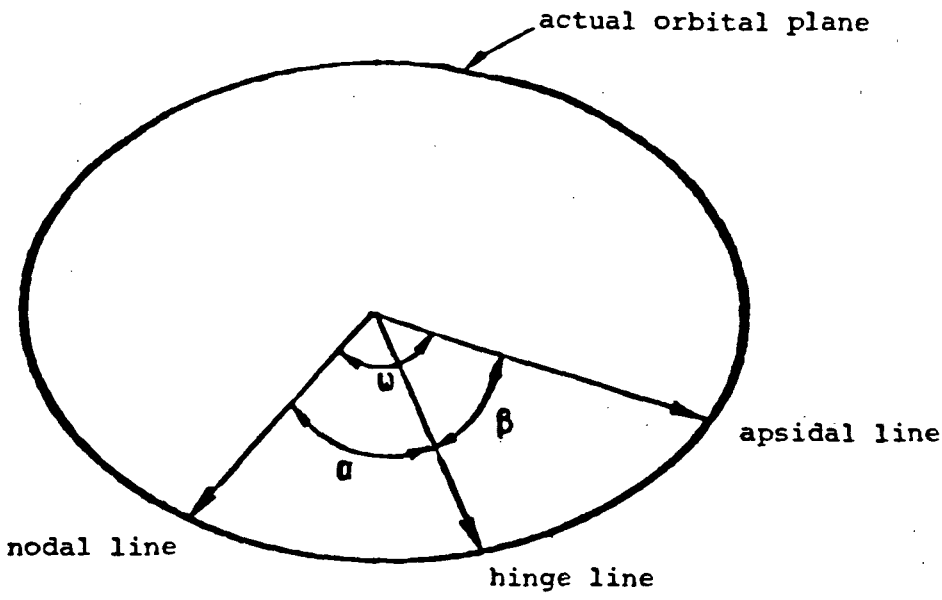


Figure 2.4

Geometric Definition of Alpha

position and velocity vectors. The hinge line vector is formed by taking the vector product of the actual angular momentum vector and the angular momentum vector of the desired orbital plane. Finally, β is found by using equation 2.27.

A subroutine has been written which calculates the orbital elements and the control parameters from position vector and velocity vector measurements. The source code for this subroutine, called ORBITS4A, is given in Appendix A.

2.4 The Aerobraking Guidance Law

The lateral guidance algorithm developed in this thesis is compatible with any aerobraking guidance law that does not specify the sign of the roll angle command. The aerobraking guidance law used to evaluate the performance of the lateral guidance algorithm is essentially the one developed in references [3,4,and 9]. The guidance output is the in-plane value of the lift to drag ratio (L/D) needed to achieve the required drag acceleration and altitude rate. The commanded in-plane L/D is obtained by modulating the direction of the lift vector. The guidance is divided into three phases: a constant attitude phase, a down control phase, and an up control phase.

The constant attitude phase and the down control phase are described in references [4 and 9]. The constant altitude phase keeps the direction of the lift vector constant (ie. constant roll angle) until the total acceleration due to the aerodynamic forces exceeds 0.05 g's, when the down control phase begins. The constant roll angle chosen for the evaluation of the lateral guidance algorithm is 90 degrees, since a lift vector which is completely out-of-plane generates the biggest possible initial plane error. A large initial plane error is desired to evaluate the performance of the lateral guidance algorithm under the worst possible operating conditions.

The down control phase modulates the lift vector to achieve a penetrating trajectory with a constant altitude rate. This type of trajectory is called an equilibrium glide trajectory. Associated with the equilibrium glide trajectory is a reference L/D, a reference drag acceleration profile, and a reference altitude rate profile. Once the equilibrium glide trajectory is achieved, the guidance effectively controls to a reference drag acceleration profile.

The commanded in-plane L/D required to attain the equilibrium glide condition is equal to the reference L/D plus correction terms based on the drag acceleration error and the altitude rate error. The drag acceleration error is the difference between the actual drag acceleration measured by the accelerometers and the calculated reference drag acceleration. A derived altitude rate calculated from the drag acceleration measurements is defined in reference [9], since measurements of the OTV current altitude are assumed to be unavailable in reference [9]. The altitude rate error used in reference [9] is then the difference between the derived altitude rate and the reference altitude rate. Unfortunately, the equation for the derived altitude rate is highly inaccurate in the presence of short term density disturbance (reference 10) which results in poor performance of the aerobraking guidance law. Therefore, for the performance evaluation of the lateral guidance algorithm, altitude rate measurements are assumed to be available from navi-

gation and the altitude rate error is now the difference between the actual altitude rate and the reference altitude rate.

The down control phase ends and the up control phase starts when the OTV velocity is within 5500 ft/sec of the desired exit velocity. The up control phase modulates the direction of the lift vector to achieve an exit trajectory which maintains a reference constant altitude rate (see reference 3). The reference constant altitude rate required to achieve the desired exit velocity is calculated from the present drag acceleration at every guidance cycle. A reference in-plane L/D needed to have a constant altitude rate is also calculated every guidance cycle. The up control phase controls to the reference in-plane L/D with feedback on the altitude rate error. The altitude rate error is the difference between the actual altitude rate and the reference constant altitude rate.

The down control phase and the up control phase are both sensitive to drag acceleration measurements. The commanded in-plane L/D for the down control phase and the reference constant altitude rate for the up control phase both depend on drag acceleration measurements. The dependency on drag acceleration measurements causes poor performance in the presence of short term density disturbances (see reference 10). Therefore, the aerobraking guidance logic is modified to include a low

pass filter on the drag acceleration measurements which improves the performance of the up control phase and the down control phase.

The aerobraking guidance law of references [3,4, and 9] is chosen to evaluate the performance of the lateral guidance algorithm, because the most severe possible conditions for controlling the position of the hinge line are provided. The position of the hinge line is only effected by the lift acceleration component normal to the current orbital plane (ie. out-of-plane), see equation 2.16, but a common characteristic of the up control phase is that the lift vector has a small out-of-plane component. Therefore, the control authority available to move the hinge line is extremely limited during the up control phase.

2.5 The Effects Of β On The On-Board Targeting Algorithm

The desired orbit for the OTV can not be exactly obtained with the aerobraking maneuver due to atmospheric density disturbances, navigation errors, uncertainties in the OTV's aerodynamics, and other guidance errors. The OTV must perform propulsive thrust maneuvers after leaving the atmosphere to achieve the desired orbit. The guidance logic which determines how to perform the propulsive thrust maneuvers is called the on-board targeting algorithm.

Significant reductions in the burn requirements and a simplification of the on-board targeting algorithm are obtained when β equals zero as discussed in Section 1.2. An on-board targeting algorithm is developed based on the assumption that β equals zero. The desired orbit is obtained with three separate burns: a relatively large perigee-raising burn, a circularization trim burn, and a plane correction trim burn. The burn sequence algorithm attempts to correct the total plane error by performing a dog-leg maneuver on the perigee-raising burn. However, β can not realistically be controlled to zero because of the constraints on the lateral guidance algorithm. Any residual plane error caused by assuming β to be zero is corrected with a trim burn. The total burn magnitude needed to achieve the desired orbit is denoted by ΔV_{approx}.

For comparison purposes, the mimimum burn magnitude (ΔV_{opt}) to achieve the desired orbit is calculated. The minimum burn magnitude is obtained by only correcting a portion of the plane error with a dog-leg maneuver and correcting the remaining plane error with a plane correction trim burn. An iteration process is used to find the portion of the plane error correction to make with the dog-leg maneuver (see Section 1.2).

The lateral guidance algorithm is designed to keep β within a certain range. This range is selected to produce the most satisfactory performance of the burn sequence algorithm while limiting the number of commanded roll reversals. The amount of the plane error which can be corrected with a dog-leg maneuver varies with the magnitude of β. The burn sequence algorithm's performance for different values of β is given in Table 2.1. The magnitude of β must be less than 2 degrees to correct at least 95% of the plane error with a dog-leg maneuver. The portion of the plane error which can be corrected with a dog-leg maneuver decreases as the magnitude of β increases. The difference between ΔV_{approx} and ΔV_{opt} is insignificant for the range of plane errors encountered until β exceeds 60 degrees. Therefore, the performance of the on-board targeting algorithm is not severely degraded by assuming β to be zero.

A subroutine called GCH.BURNS4A has been wriiten by Tom Fill of the Charles Stark Draper Laboratory which calculates the required burn mag-

nitudes needed to achieve the desired orbit. This subroutine has been modified to calculate also ΔV_{approx}.

Table 2.1
On-board Targeting Algorithm Performance For Different Values Of β

β (deg's)	plane error (deg's)	percentage of plane error corrected[1]	ΔV_{approx} (ft/sec)	ΔV_{opt} (ft/sec)
.043	.025	99.9%	200.476	200.507
.694	.025	98.8%	198.163	198.173
1.96	.061	96.5%	204.435	204.425
4.99	.055	91.3%	226.796	226.772
9.36	.058	83.6%	213.426	213.334
20.7	.069	64.4%	250.580	250.072
42.8	.049	31.7%	211.191	209.490
60.8	.026	12.7%	212.413	210.522
92.0	.048	.06%	231.421	222.082

[1]with a dog-leg maneuver on the perigee raising burn.

CHAPTER 3

LATERAL GUIDANCE ALGORITHM DESIGN

The goal of the lateral guidance algorithm developed in this thesis is to minimize β by the time the OTV leaves the atmosphere without using an excessive number of roll reversals. If β is near zero, the solution provided by the on-board targeting algorithm is close to the optimal solution and the majority of the plane error is corrected at little cost with a dog-leg maneuver. Section 3.1 gives a general overview of the lateral guidance algorithm. The remainder of Chapter 3 discusses the development of the different phases of the lateral guidance algorithm in detail. A subroutine has been written to implement the lateral guidance algorithm described in this chapter. The source code for the subroutine, called GCH.GUID8C, is given in Appendix A.

3.1 Overview Of The Lateral Guidance Algorithm

The lateral guidance algorithm nominally consists of four different phases unless the desired plane error ($\delta_{desired}$) falls beneath a certain value. The first and second phases regulate the plane error about zero. An additional phase is inserted between the second and third phases when the desired plane error, which is defined in Section 2.2, is less than 0.01 degrees. This additional phase is essentially a modified version of the second phase. However, instead of trying to null the plane error, the plane error is driven to $\delta_{desired}$. The third phase prevents the hinge line and radius vector from coinciding (ie. η equals zero). The fourth phase restricts β to a range about zero.

3.2 Plane Error Control

Controlling the size of the plane error is more advantageous than controlling the hinge line position during the first two phases of the lateral guidance algorithm. The hinge line position varies rapidly during the initial stage of the aerobraking maneuver due to the large out-of-plane lift forces and small plane error (see equation 2.16). This rapid variation of hinge line position makes it impossible to provide fine control of β without commanding roll reversals at short intervals, which is undesirable. However, the plane error can be easily controlled with long intervals between roll reversals. The size of the plane error is important, because the rate of change of the hinge line position is inversely proportional to the plane error magnitude. Controlling the plane error early in the trajectory insures the ability to control β latter in the trajectory. Furthermore, by keeping the plane error small, the burn magnitude needed to correct the plane error is prevented from becoming large. Thus, in the first two phases, priority is placed on nulling the plane error.

The first two phases are essentially the lateral guidance algorithm developed in reference [6]. The size of the plane error is controlled by minimizing or zeroing the velocity out-of-plane error (θ_V). A phase plane deadband is defined in which no control action is taken as long as

θ_V is within the deadband. A roll reversal is commanded when the value of θ_V exceeds the deadband limits.

The deadband limits are ±0.5 degrees during the first phase. When the velocity of the OTV is within 1600 ft/sec of the desired exit velocity, the second phase begins and the deadbands are reduced to ±0.05 degrees. Finer control is maintained during the second phase, since the out-of-plane lift forces available to correct the plane error have decreased. If the out-of-plane lift forces become too small during the first phase, the deadband limits are changed to ±0.25 degrees. This is necessary to prevent a large plane error from forming during periods of reduced control authority. For both phases, the deadband limits are slightly biased to compensate for the effects of the gravity component normal to the desired plane. A flag is set to prevent unnecessary roll reversals when θ_V is outside the deadband but is moving towards zero.

3.3 Eta Control

The ability to control the hinge line position will be lost if the size of the plane error is still being regulated instead of the hinge line position when the out-of-plane lift forces (ie. the lift components normal to the current orbital plane) have fallen beneath a certain level. The third phase commences when the velocity of the OTV is within 800 ft/sec of the desired exit velocity. This gives the second phase enough time to reduce the plane error to an acceptable value. Also at this point, the out-of-plane lift forces have decreased to a level where the plane error is no longer changing rapidly. If η is close to zero, the out-of-plane lift forces are still large enough to move the hinge line away from the current OTV position, but are too large to restrict the hinge line to a small region without requiring numerous roll reversals.

The ability to vary the position of the hinge line is severely limited if the hinge line is near the current OTV position (ie. η is small), as discussed in Section 2.1. The magnitude of η must not get too small or the out-of-plane lift forces will not be sufficient to insure that β will be zero when the OTV leaves the atmosphere. On the other hand, the number of required roll reversals is reduced when the magnitude of η is small, since the rate of change of the hinge line

position is also small. Thus, the goal of the third phase is to keep η greater than some predetermined value.

An exclusion zone is defined around the current OTV position. A flag is set to prevent unnecessary roll reversals when the hinge line is inside the exclusion zone but is moving away form the current OTV position. If the hinge line enters the exclusion zone and the flag is not set, a roll reversal is commanded. Initially, the magnitude of η is kept above 8 degrees. This value limits the number of roll reversals required during the third phase while insuring that the out-of-plane lift forces will be sufficient to move the hinge line away form the current OTV position and towards the apsidal line.

The exclusion zone is enlarged when the rate of change of the longitude of the hinge line ($d\Psi/dt$) is less than 1.5 degrees/sec. The magnitude of η is now kept above 48° if β is greater than the true anomaly (ie. η is positive) or above 24 degrees if β is less than the true anomaly (ie. η is negative). The enlargement of the exclusion zone is needed to keep the distance between the hinge line and the apsidal line from getting too large when the ability to move the hinge line is limited. The limits of the exclusion zone are unsymmetric because the relative distance of the hinge line from the apsidal line depends on the sign of η. The apsidal line is usually close to the current OTV position during the third phase. As a consequence, the distance along the path between

the hinge line and the apsidal line which avoids the exclusion zone is generally much larger when η is positive than when η is negative.

The logic to decide when to perform a roll reversal must be modified when η is negative during the latter stages of the third phase. The apsidal line and the current OTV position are constantly moving apart during the third phase. This movement eventually invalidates the assertion made about the relative distance between the hinge line and the apsidal line based on the sign of η. Another problem is caused by the rapid movement of the apsidal line away from the current OTV position during the latter stages of the third phase. If β is negative, the magnitude of β will decrease due to the movement of the apsidal line. However, if β is positive, the magnitude of β will increase which is very undesirable. If η is negative, $d\Psi/dt$ is less than 0.3 degrees/sec, and β is greater than 3 degrees, a roll reversal is commanded. This logic prevents the hinge line from getting too far away from the apsidal line when the ability to move the hinge line is limited and the apsidal line is rapidly moving away from the hinge line.

3.4 Beta Control

The magnitude of β must be regulated before the out-of-plane lift forces become too small to move the hinge line away from the current OTV position and reduce β to zero. Phase three ends and phase four begins when the measured out-of-plane lift acceleration is less than 0.8 ft/sec^2. If phase three remains in control beyond this point, the out-of-plane lift forces will not be large enough to null β when β is large (ie. the distance between the apsidal line and the hinge line is large). When phase four starts, the out-of-plane lift forces are still large enough to move the hinge line away from the current OTV position and drive β to zero regardless of the initial size of β. However, the out-of-plane lift forces have decreased enough by this time to confine the hinge line within a region about the apsidal line without requiring numerous roll reversals.

The phase four control strategy is to keep β within a phase plane deadband. A roll reversal is commanded if β is outside the deadband. A flag is set to prohibit unnecessary roll reversals when β is outside the deadband but the hinge line is moving towards the apsidal line. The deadband limits depend on the magnitude of the rate of change of the longitude of the hinge line ($d\Psi/dt$). This quantity was chosen as the basis for the deadband limits because it reflects the effects of both the plane error magnitude and the out-of-plane lift forces on the hinge

line position. When the out-of-plane lift forces are small, they are no longer the dominant influence on the hinge line position. Both the out-of-plane lift forces and the plane error magnitude equally affect the rate of change of the hinge line position (see equation 2.16) during the fourth phase. Therefore, the selection of the deadband limits must take into account the plane error magnitude and the amount of out-of-plane lift forces.

Initially, the deadband limits are -20 degrees and 3 degrees. The deadband limits are cut in half to -10 degrees and 1.5 degrees when $d\Psi/dt$ is less than 0.3 degrees/sec. Finally, the deadband limits are further reduced to -2 degrees and 1 degree when $d\Psi/dt$ is less than 0.15 degrees/sec. The shrinking size of the deadband reflects the desire to limit the number of roll reversals as much as possible while still insuring that β will be near zero when the OTV leaves the atmosphere.

The deadband is asymmetric to compensate for the movement of the apsidal line. During the early stages of the fourth phase, the apsidal line is moving rapidly away from the current OTV position. The rate of change of the apsidal line is effected by the in-plane aerodynamic forces as well as the out-of-plane aerodynamic forces. For this reason, the rate of change of the apsidal line is much greater than the rate of change of the hinge line. The movement of the apsidal line causes the magnitude of β to decrease when β is negative, since the apsidal line

will be moving towards the hinge line. However, the magnitude of β rapidly increases when β is positive because the apsidal line will be moving away from the hinge line. The upper limits on β must be kept small to prevent the hinge line from getting too far away from the apsidal line when the apsidal line is rapidly moving away from the hinge line and the ability to move the hinge line is decreasing. Conversely, the lower limits can be larger since the distance between the hinge line and the apsidal line decreases rapidly due to the movement of the apsidal line.

There exists a potentially dangerous situation during the fourth phase. If η is positive and the hinge line is moving towards the apsidal line, a roll reversal will not be commanded. This is extremely undesirable, because the value of η will decrease and thus the ability to move the hinge line will diminish. The fourth phase must be modified to prevent this. The value of α was originally defined to indicate the shortest distance between the hinge line and the apsidal line. Unfortunately, the shortest distance goes through the current OTV position under certain circumstances. The value of α is redefined to avoid this situation. If η is positive, the value of α is redefined to be:

$$\alpha = \alpha - 180^\circ \qquad (3.1)$$

which puts α in the third quadrant. The redefined value of α represents the distance between the apsidal line and the hinge line which does not pass through the current OTV position. This modification causes the lateral guidance algorithm to command a roll reversal, since the value of α will now indicate that the hinge line is moving away from the apsidal line.

3.5 Small Plane Error Control

Controlling the hinge line position and the size of β is no longer an immediate concern when the desired plane error is less than 0.01 degrees. The burn magnitude needed to correct a plane error of that magnitude or smaller is insignificant. Also, the number of required roll reversals will be excessive in order to drive β to zero when the desired plane error is less than 0.01 degrees. A modified version of the second phase is implemented if the desired plane error is less than 0.01 degrees during the third phase. If the desired plane error is still less than 0.01 degrees and $d\Psi/dt$ is greater than 1.5 degrees/sec during the fourth phase, phase three is selected; otherwise, the fourth phase remains in control if $d\Psi/dt$ is less than 1.5 degrees/sec. However, if the desired plane error falls below 0.001 degrees, the number of roll reversals commanded will be excessive even for the eta control phase logic. As a result, if the desired plane error is less than 0.001 degrees and $d\Psi/dt$ is greater than 0.2 degrees/sec during the fourth phase, the modified version of the second phase is implemented; otherwise, the fourth phase remains in control if $d\Psi/dt$ is less than 0.2 degrees/sec.

The modified version of the second phase controls the plane error magnitude by keeping θ_V within a certain range of $\theta_{V_{desired}}$ (see Section 2.2). A phase plane deadband is defined and no control action is taken

as long as Θ_V stays inside the deadband. The deadband limits are still ±0.05 degrees but are now biased by the value of $\Theta_{V\ desired}$. A roll reversal is commanded if Θ_V exceeds the deadband limits. A flag is set to prevent unnecessary roll reversals when Θ_V is outside the deadband and the difference between Θ_V and $\Theta_{V\ desired}$ is decreasing.

The measured value of the true anomaly used to calculate $\Theta_{V\ desired}$ is biased by 20 degrees until the actual true anomaly exceeds 40 degrees. This biasing is done to take into account the varying apsidal line position which is moving further away form the current OTV position. The sign on $\Theta_{V\ desired}$ is not determined by equation 2.26. For convenience, the sign on $\Theta_{V\ desired}$ is chosen to be the same as the current sign of Θ_V. When the magnitude of $\Theta_{V\ desired}$ is less than 0.05 degrees, this sign convention effectively enlarges the deadband. The enlargement of the deadband is desirable for small $\Theta_{V\ desired}$, because the number of roll reversals required is reduced without increasing the difference between Θ_V and $\Theta_{V\ desired}$.

CHAPTER 4

LATERAL GUIDANCE ALGORITHM EVALUATION

4.1 Aerobraking Simulator

A computer simulation was developed to test and evaluate the performance of the lateral guidance algorithm described in Chapter 3. This computer simulation consists of several subprograms. The four major subprograms are described below. The computer codes for the subprograms are presented in Appendix A.

4.1.1 Driver Subprogram

The driver subprogram performs all the input and initialization operations. The initial actual state and the initial navigated state of the OTV are computed based on the inputs provided by the user. The addition of navigation errors and/or trajectory perturbations to the

initial state of the OTV, aerodynamic properties (ie. ballistic coefficient and lift to drag ratio), and the addition of density disturbances are all performed by this subprogram.

4.1.2 Environment And Navigation Subprogram

The environment and navigation subprograms handle the actual simulation of the OTV flight trajectory and perform the navigation functions. The environment section computes the actual current state of the OTV and propagates the actual flight trajectory of the OTV. The navigation section computes the current navigated state of the OTV and propagates the navigated flight trajectory of the OTV. The navigation section also computes the altitude rate based on the navigated velocity and flight path angle. This subprogram executes the guidance and control subprograms and performs the output operations.

4.1.3 Guidance Subprogram

The guidance subprogram contains the code for the aerobraking guidance law of references [3 and 4] and the lateral guidance algorithm described in Chapter 3. The inputs to the guidance subprogram are the navigated velocity, altitude, altitude rate, and the accelerometer measurements. The lift and drag acceleration components used by the guidance subprogram are computed from the accelerometer measurements. The

output of the guidance subprogram is the magnitude and sign of the commanded roll angle.

4.1.4 Control Subprogram

The control subprogram executes the maneuver needed to attain the commanded roll angle. The maximum allowable roll rate and roll acceleration are taken into account by the control subprogram. As a consequence, the desired roll angle may not be attained immediately.

4.2 Vehicle Characteristics And Testing Methodology

The aerodynamic characteristics of the OTV are essential in determining the roll angle history for a particular trajectory. The OTV used in evaluating the lateral guidance algorithm has a lift to drag ratio of 0.3 and a ballistic coefficient of 10 lbs/ft^2. To simplify the interpretation of the simulation test results, the maximum roll rate and roll acceleration are assumed to be 1000 degrees/sec and 1000 $degrees/sec^2$ respectively. The unrealistically large values for the roll rate and the roll acceleration insure that the commanded roll angle will be achieved immediately.

The aerobraking guidance law used to test the lateral guidance algorithm is designed to control the OTV for a geosynchronous return mission. In a geosynchronous return mission, the OTV is transferring from a geosynchronous orbit to a low Earth orbit. Normally, it is desired that the OTV will rendezvous with the shuttle. Thus, the desired post-aerobraking target orbit is a circular orbit 150 nautical miles above the surface of the Earth at an inclination of 28.5 degrees with the equatorial plane. Furthermore, the longitude of the ascending node for the desired orbit is 0.0 degrees.

Numerous simulations are made under different operating conditions to fully evaluate the performance and the advantage of the hinge line

lateral guidance algorithm described in Chapter 3. The OTV enters the atmosphere from a geosynchronous orbit with a certain vacuum perigee. The vacuum perigee is the perigee that the orbit would have if the Earth had no atmosphere. The density profile which the OTV encounters during the atmospheric flight is changed by varying the vacuum perigee. This new density profile generates a new commanded roll angle history.

Another way to generate different density profiles is to run the vacuum perigee dispersion cases with thick atmospheres. A thick atmosphere means that the nominal density (as obtained from the standard U.S. 1962 Atmosphere Model) is increased by a constant factor. The thick atmosphere not only generates a new commanded roll angle history, but also increases the aerodynamic forces generated during the aerobraking maneuver. The greater aerodynamic forces increase the rate of change of the plane error magnitude, the apsidal line position, and the hinge line position.

The presence of a thick atmosphere stresses both the hinge line lateral guidance algorithm and the plane error lateral guidance algorithm of reference [6]. The increase in the rate of change of the apsidal line position provides a difficult test for the hinge line lateral guidance algorithm. This algorithm is trying to drive the hinge line to the current apsidal line position which is now moving over a larger distance and at a faster rate. The thick atmosphere also degrades the perform-

ance of the plane error lateral guidance algorithm which just controls the plane error magnitude. The plane error magnitude increases more rapidly in a thick atmosphere because the out-of-plane lift forces generated are larger than those generated in the nominal atmosphere. Thus, more roll reversals are required to keep the plane error magnitude within the deadband and the final plane error is more likely to have a larger magnitude in a thick atmosphere than in the nominal atmosphere.

Another density variation which might affect the performance of the lateral guidance algorithm is the pot-hole density disturbance. A pot-hole density disturbance is a sudden decrease in the actual density from the nominal density over a short period of time (see Figure 4.1). The length and time of occurrence of the pot-hole are based on the OTV velocity. The nominal density is decreased by a constant factor (RHOBIAS) when the OTV velocity is within a certain value (VELBIAS2) of the desired exit velocity. The nominal density is used again for the remaining flight trajectory when the OTV velocity is within a smaller value (VELBIAS1) of the derived exit velocity. The values of VELBIAS2 and VELBIAS1 are chosen to place the pot-hole towards the end of the first phase and before the start of the second phase of the lateral guidance algorithm. This placement of the pot-hole increases the initial size of the velocity and position components normal to the desired orbital plane (ie. out-of-plane errors) at the start of the second phase.

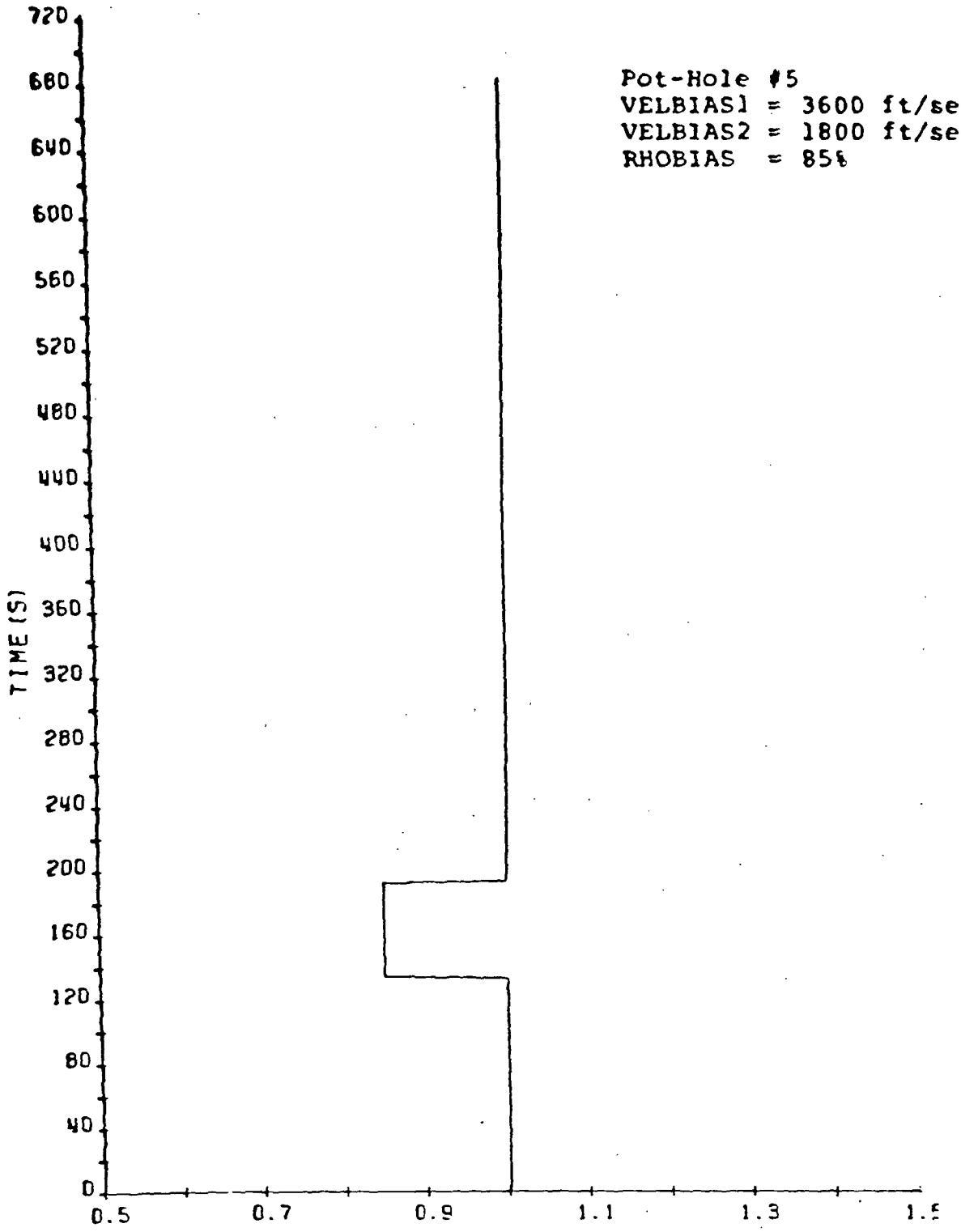


Figure 4.1
Typical Density Profile With A Pot-Hole Present

Several simulations with pot-hole density disturbances of different length were made to examine the performance of the lateral guidance algorithm in the presence of large velocity and position out-of-plane errors at the start of the second phase. The pot-holes provided the most severe test for the hinge line lateral guidance algorithm, whereas they had a negligible effect on the performance of the plane error lateral guidance algorithm. The large velocity and position out-of-plane errors present at the start of the second phase will eventually be reduced by the plane error lateral guidance algorithm, since the magnitude of the velocity out-of-plane error is being regulated during the entire flight. However, the hinge line lateral guidance algorithm will not reduce these errors by the same degree, since the hinge line position is being controlled instead of the velocity out-of-plane error. The large position out-of-plane error produces a large desired plane error (see equation 2.25). The larger plane error not only reduces the ability to move the hinge line, but more importantly, increases the burn magnitude needed to correct the plane error.

Several simulations with pot-hole density disturbances combined with a thick atmosphere were made to further evaluate the performance of the lateral guidance algorithm. The thick atmosphere degrades the performance of the plane error lateral guidance algorithm, as discussed previously, but improves the performance of the hinge line lateral guidance algorithm. The out-of-plane lift forces available to correct the plane

error and move the hinge line are increased by the thick atmosphere. As a result, the velocity and position out-of-plane errors will be smaller at the start of the eta control phase than in the pot-hole cases for a nominal atmosphere. This reduces the desired plane error and the final plane error in the pot-hole cases with a thick atmosphere. The smaller plane error reduces the burn magnitude needed to correct the plane error.

4.3 Performance Evaluation

The advantage of controlling the hinge line position instead of the plane error magnitude is demonstrated by making numerous simulation runs under various operating conditions. For each operating condition, two simulation runs are made with each one using a different option for the lateral guidance algorithm. Under option one, the plane error lateral guidance algorithm of reference [6] is used. Under option two, the hinge line lateral guidance algorithm described in Chapter 3 is used.

The performance of the lateral guidance algorithm can be evaluated by several different parameters. The most useful parameters in determining whether option two is more advantageous than option one is the total burn magnitude needed to place the OTV in the desired orbit. The burn magnitude needed to place the OTV in the desired circular orbit will be the same for both option one and option two. However, the burn magnitude needed to correct the plane error will be different for each option, since each option uses a different approach in minimizing the burn magnitude needed to correct the plane error. Option two controls the location of the hinge line during the latter stages of the flight trajectory, while option one controls the plane error magnitude throughout the entire flight trajectory. As a result, the final plane error when option two is used could be larger than the final plane error when option one is used. The increase in the burn magnitude needed to cor-

rect the larger plane error is offset by the savings made when the hinge line and apsidal line coincide (ie. β equals zero). By comparing the total burn magnitudes from option one and option two, the advantage of using option two is shown.

An important performance variable for the hinge line lateral guidance algorithm (option two) is the angle between the hinge line and apsidal line (β). The magnitude of β determines the portion of the plane error which can be corrected by performing a dog-leg maneuver on the perigee-raising burn. The plane error can be corrected completely by the dog-leg maneuver when β is zero. The on-board targeting algorithm associated with option two assumes that β equals zero and tries to correct the plane error entirely with a dog-leg maneuver. Any residual plane error left after performing the dog-leg maneuver is corrected with the trim burn. The burn magnitude needed to correct a particular plane error decreases as the magnitude of β decreases. Thus, the total burn magnitude is minimized when β is zero.

The plane error magnitude is an important performance variable for the plane error lateral guidance algorithm (option one). Option one tries to minimize the total burn magnitude by keeping the velocity out-of-plane error within a deadband. The on-board targeting algorithm associated with option one uses two different methods for determining the total burn magnitude. The first method solves for the required

burns without using a dog-leg maneuver. The plane error is corected completely with the trim burn. The total burn magnitude found using the first method represents the maximum total burn magnitude needed to achieve the desired orbit given a particular set of post-aerobraking trajectory conditions. The second method uses a dog-leg maneuver to find the minimum total burn magnitude required to achieve the desired orbit. However, the portion of the plane error to correct with the dog-leg maneuver is not obvious, since option one does not control the hinge line position. As a consequence, the magnitude of β could be of any size. An iteration process is used to determine the portions of the plane error to correct with the dog-leg maneuver and the trim burn which will minimize the total burn magnitude. As β approaches 90 degrees, the difference between the maximum and the minimum total burn magnitude approaches zero.

The advantage of option two over option one can be seen by comparing the total burn magnitudes obtained from each option under the same operating conditions. The difference between the maximum total burn magnitude from option one and the total burn magnitude from option two is denoted by ΔV_{max}. The difference between the minimum total burn magnitude from option one and the total burn magnitude from option two is denoted by ΔV_{min}. Thus, ΔV_{max} and ΔV_{min} represent the fuel savings or fuel penalty incurred by using option two instead of option one. When ΔV_{max} and/or ΔV_{min} are positive, the total burn magnitude of option two

is smaller than the associated total burn magnitude of option one. When ΔV_{max} and/or ΔV_{min} are negative, the total burn magnitude of option two is larger than the associated total burn magnitude of option one.

Another important quantity in evaluating the advantage of option two over option one is the number of commanded roll reversals. It is desirable to minimize the number of required roll reversals due to roll jet fuel consumption and structural considerations. The fuel savings made by using the hinge line lateral guidance algorithm (option two) could be negated if the total number of commanded roll reversals is significantly greater than the number commanded when the plane error lateral guidance algorithm (option one) is used.

4.4 Simulation Test Results

The lateral guidance algorithm is evaluated over a wide variety of operating conditions as discussed in Section 4.2. The results from the simulation runs are presented in tables in the following subsections. Figures with plotted data from some of the test runs will be presented only when they contain some new information.

4.4.1 Perigee Dispersion Cases

The hinge line lateral guidance algorithm (option two) has better performance than the plane error lateral guidance algorithm (option one) over a wide range of vacuum perigees. The results of the simulation runs are presented in Table 4.1. In all the cases, the total burn magnitude of option two was smaller than both total burn magnitudes of option one.

The difference between ΔV_{max} and ΔV_{min} was greater than 10 ft/sec in only three cases (1, 6, and 10). This large difference was due to the chance occurrence that the final magnitude of β obtained under option one was small. The small magnitude of β enabled a large portion of the plane error to be corrected with a dog/leg maneuver which greatly reduced the minimum total burn magnitude of option one in these three cases. Despite this reduction, the total burn magnitude of option two

was still smaller than the minimum total burn magnitude of option one, since the final β of option two was smaller than the final β of option one. The smaller magnitude of β allowed a larger portion of the plane error to be corrected under option two with a dog-leg maneuver than under option one.

Table 4.1
Simulation Results For The Perigee Dispersion Cases

Case	Perigee (n.m)	Option One			Option Two			ΔV_{max} (ft/s)	ΔV_{min} (ft/s)
		β (degs)	Plane error (degs)	Roll RV.	β (degs)	Plane error (degs)	Roll RV.		
1	44.0	-8.994	.0416	5	-1.436	.0497	5	16.73	2.03
2	43.0	57.14	.0425	6	.962	.0390	9	19.15	16.39
3	42.0	51.36	.0370	6	.694	.0254	11	17.14	15.63
4	41.0	42.86	.0491	5	-1.434	.0127	11	20.74	14.39
5	40.5	-15.25	.0178	4	.364	.0220	6	7.64	1.99
6	40.0	10.68	.0638	6	-3.686	.0385	6	26.49	5.37
7	39.2	97.37	.0332	6	1.963	.0608	7	11.18	11.06
8	39.0	92.02	.0479	5	1.722	.0579	5	17.16	17.15
9	38.0	65.21	.0312	6	-1.760	.0239	10	12.60	11.41
10	37.0	31.26	.05434	7	.991	.0241	9	23.68	13.01

In all but four of the cases, the final plane error of option two was slightly smaller than the final error of option one. The difference between the final plane errors of option one and option two were not responsible for the large reductions of the total burn magnitudes of option two as compared with the total burn magnitudes of option one. The large reductions in the total burn magnitude of option two were mostly obtained by keeping β small which allowed a large portion of the plane error to be corrected with a dog-leg maneuver.

The largest ΔV_{min} was 17.15 ft/sec which represents the greatest reduction in the total burn magnitude obtained by using option two instead of option one. The largest ΔV_{max} was 26.49 ft/sec which represents the greatest reduction in the total burn magnitude obtained by using option two instead of option one, if the on-board targeting algorithm used by option one does not or can not use an iteration process to find the minimum total burn magnitude. Only in two cases did option two require more than four roll reversals than option one.

Figure 4.2 shows the commanded roll angle history of option two for case 10. The last five roll reversals are commanded by the last two phases of the hinge line lateral guidance algorithm which control the hinge line position. The first two roll reversals are commanded to keep the hinge line outside the exclusion zone. The last three are commanded to keep β inside the deadband. Figure 4.3 shows the variation of the

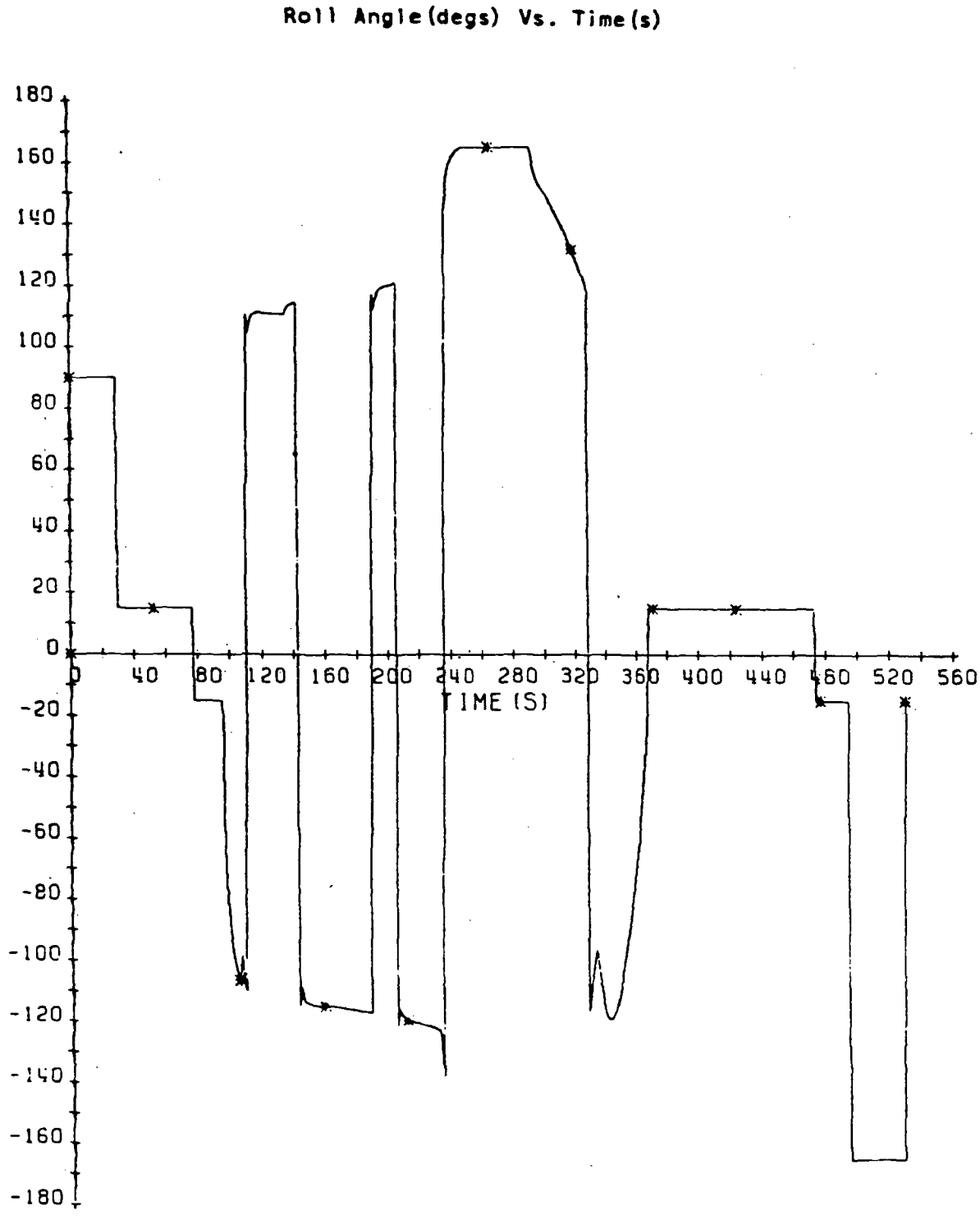


Figure 4.2
Option Two
Roll Angle Vs. Time: 37 n.m. Perigee

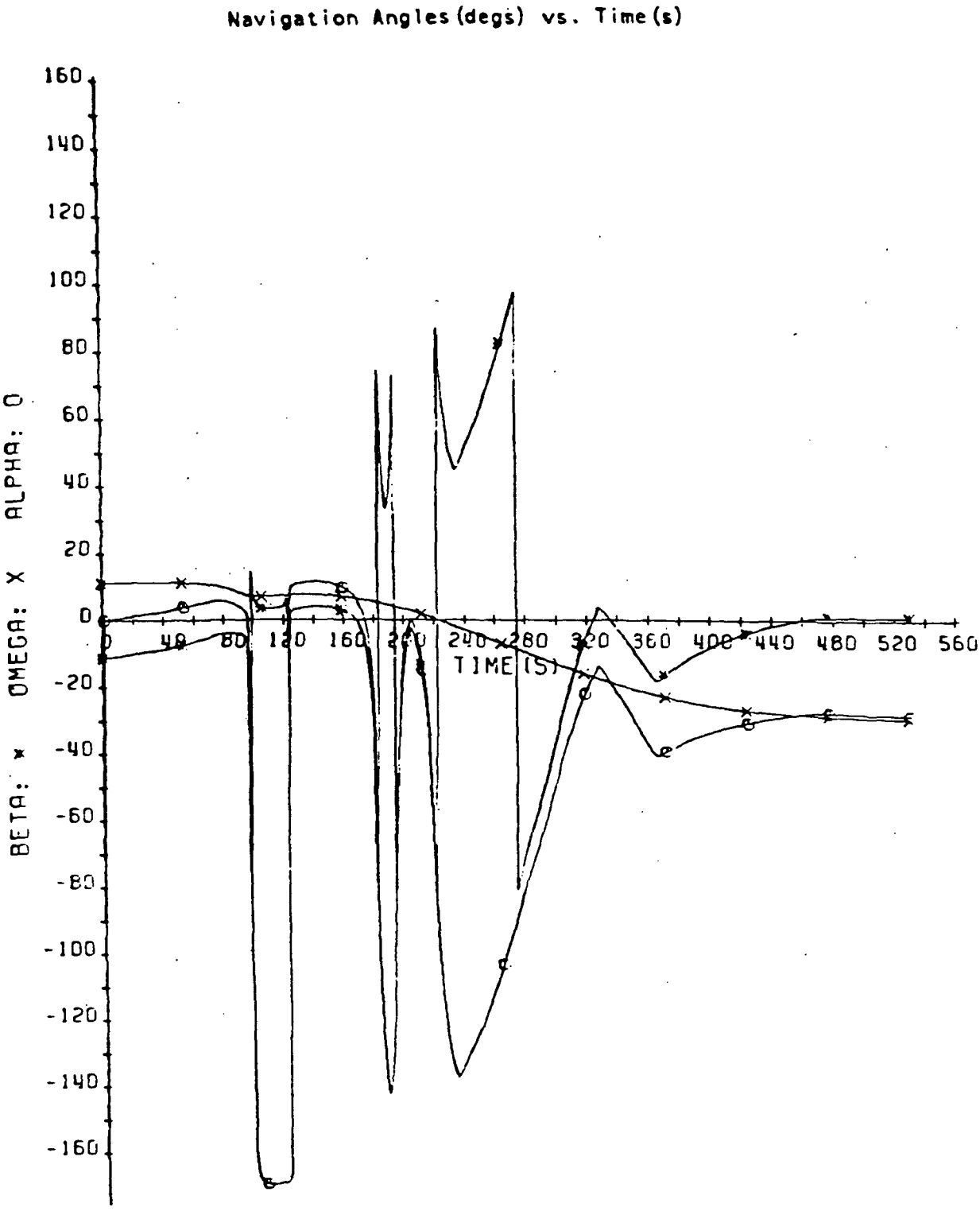


Figure 4.3
Option Two
Navigation Angles Vs. Time: 37 n.m. Perigee

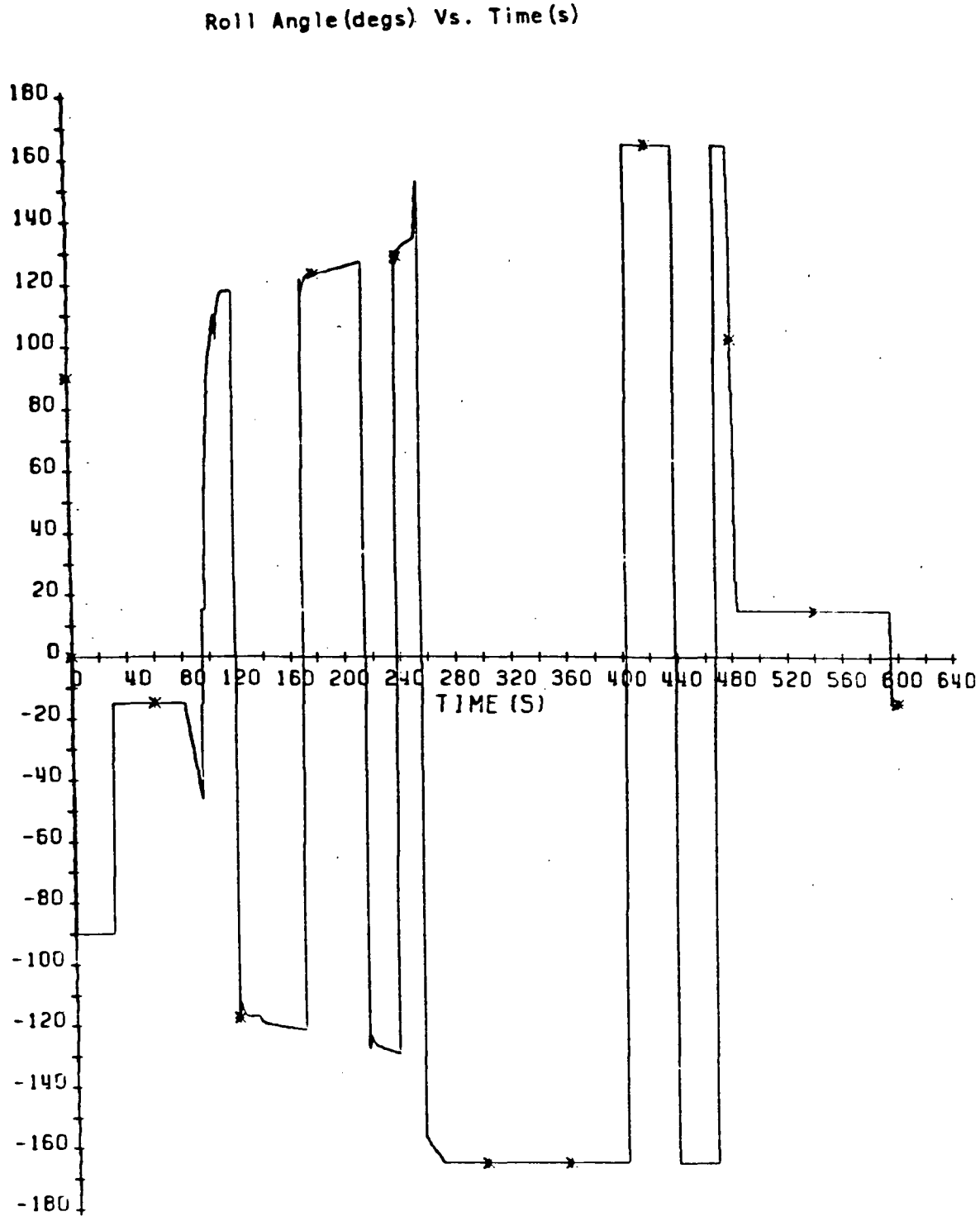


Figure 4.4
Option Two
Roll Angle Vs. Time: 38 n.m. Perigee Run

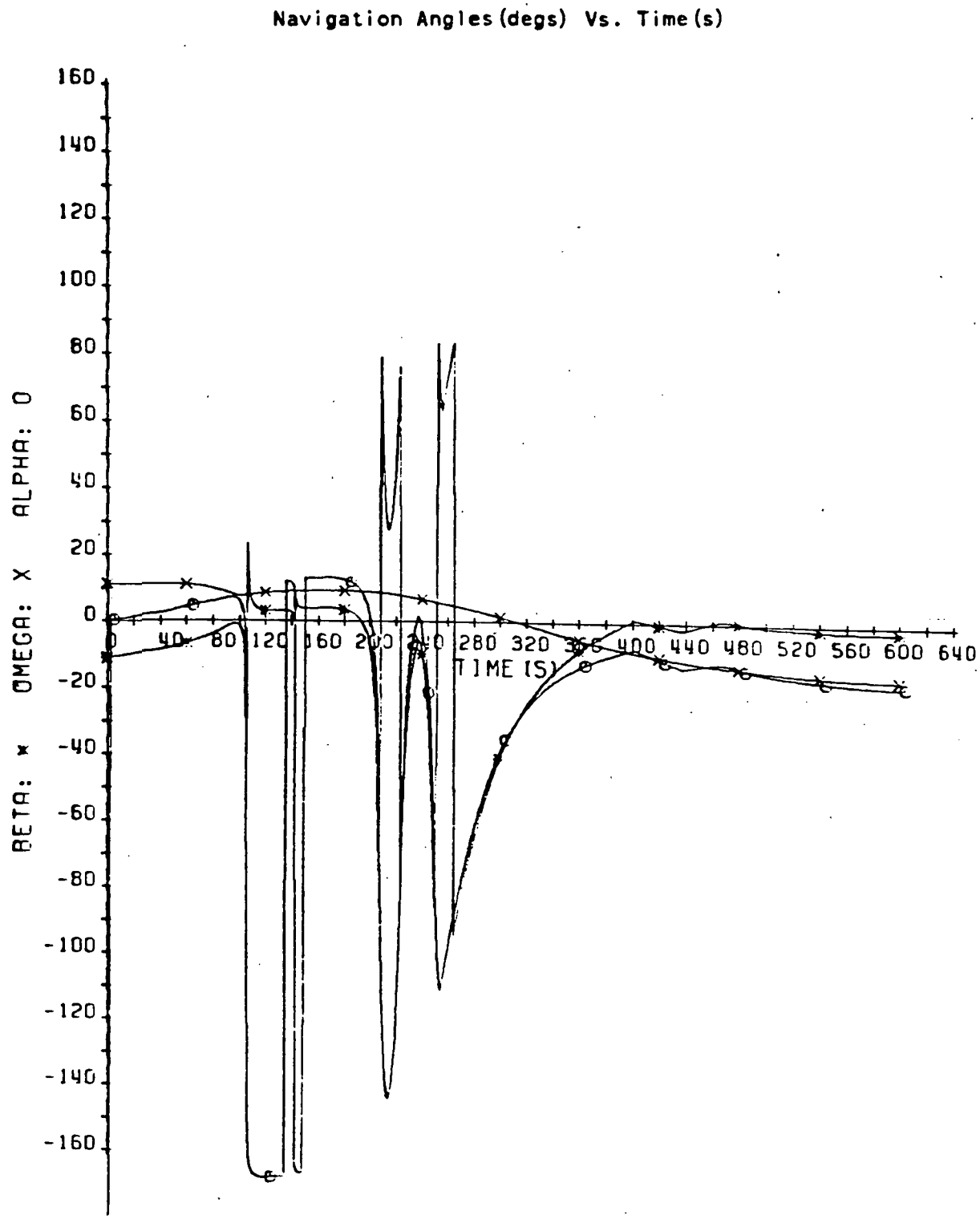


Figure 4.5
Option Two
Navigation Angles Vs. Time: 38 n.m. Perigee Run

navigation angles (β, α, and ω) of option two versus time, where ω is the argument of perigee. Figures 4.4 and 4.5 show the commanded roll angle history and the navigation angles histories of option two for case 9. The last four roll reversals are commanded to keep β inside the deadband with the smallest limits. The number of roll reversals used in case 9 could be reduced without effecting the performance of the lateral guidance algorithm by altering the criteria for changing the deadband limits. The commanded roll angle histories and the navigation angles histories in Figures 4.2 through 4.5 are typical for the majority of the simulation runs using option two for all the operating conditions made in this thesis and not just the perigee-dispersion cases.

The commanded roll angle histories and navigation angles histories of option two for cases 6 and 8 are given in Figures 4.6 through 4.9. In both these cases, the beta control phase started just in time for β to be driven to zero before the OTV left the atmosphere. In case 6, no roll reversals were commanded to keep β in the deadband (see Figures 4.6 and 4.7). In case 8, only one roll reversal was needed to keep β inside the deadband (see Figures 4.8 and 4.9). As a result, the number of roll reversals commanded by option two equaled the number commanded by option one in cases 6 and 8.

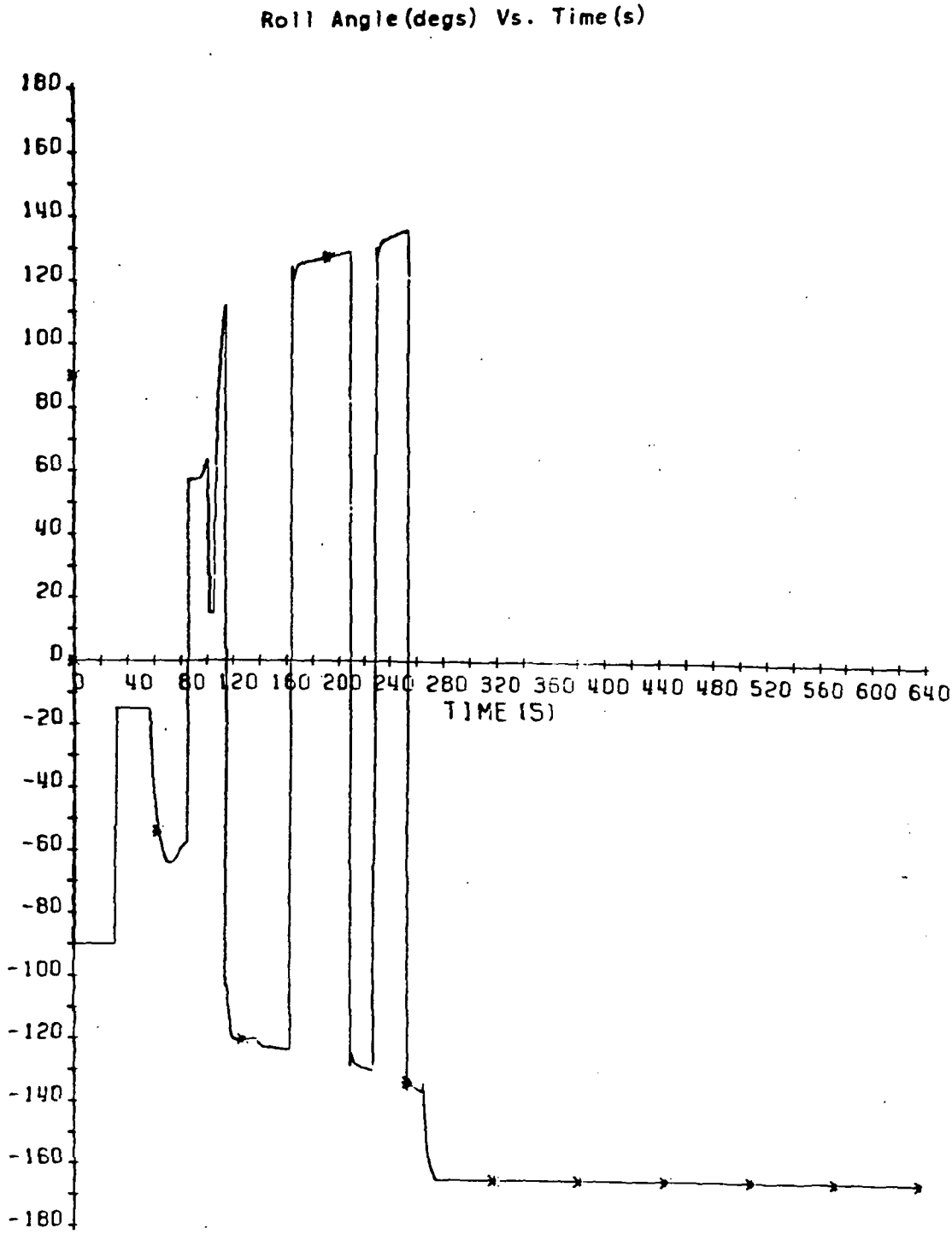


Figure 4.6
Option Two
Roll Angle Vs. Time: 40 n.m. Perigee Run

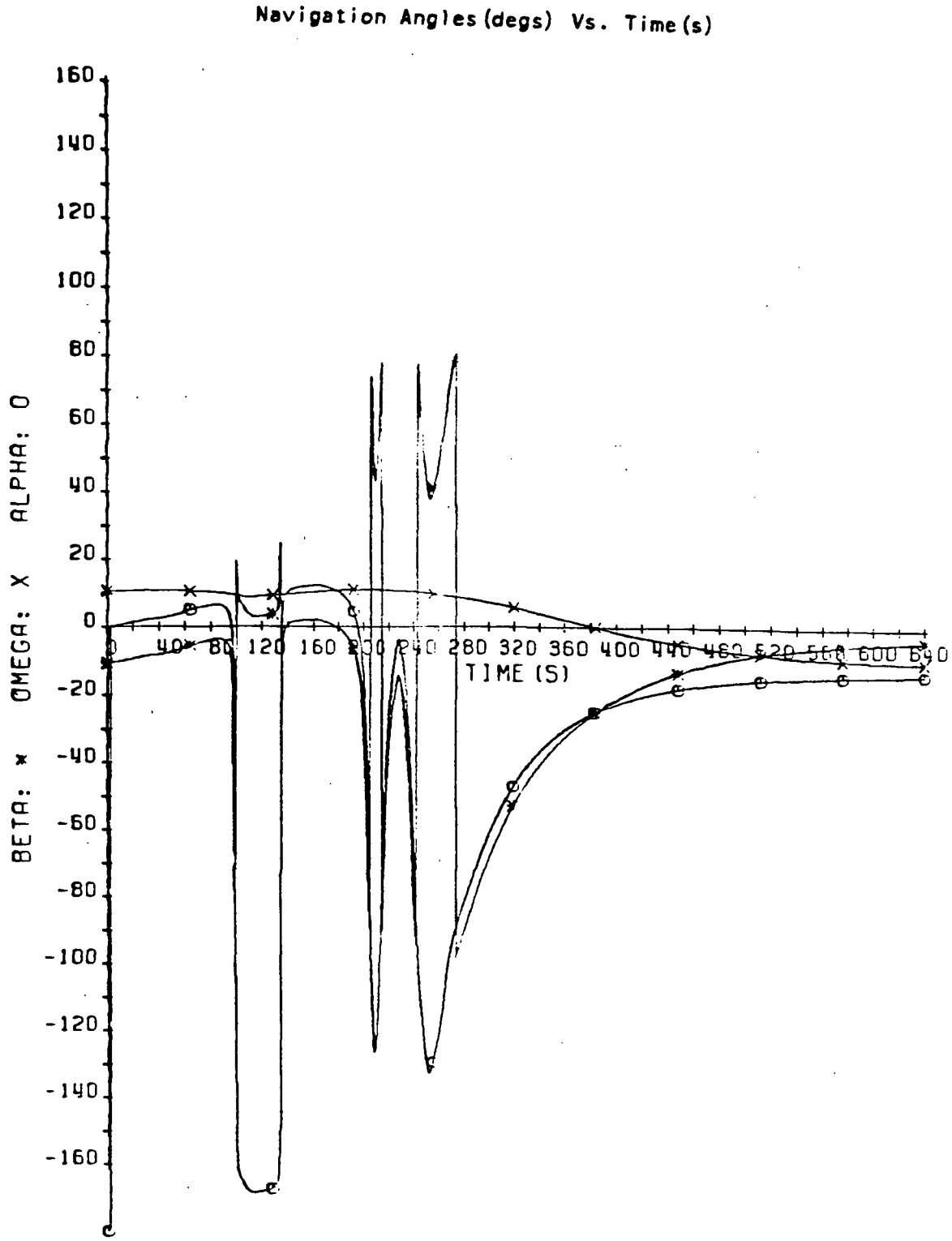


Figure 4.7
Option Two
Navigation Angles Vs. Time: 40 n.m. Perigee Run

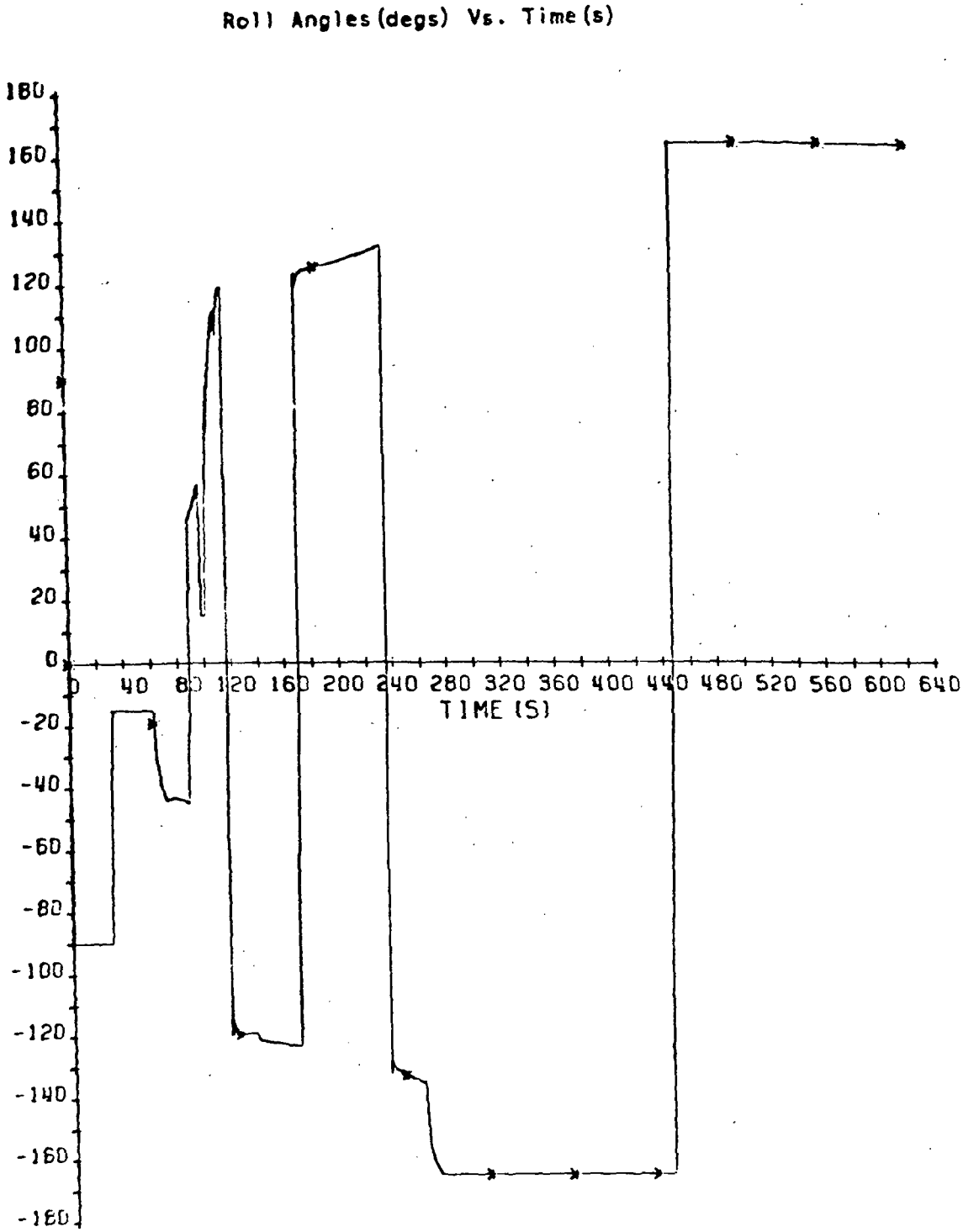


Figure 4.8
Option Two
Roll Angle Vs. Time: 39 n.m. Perigee Run

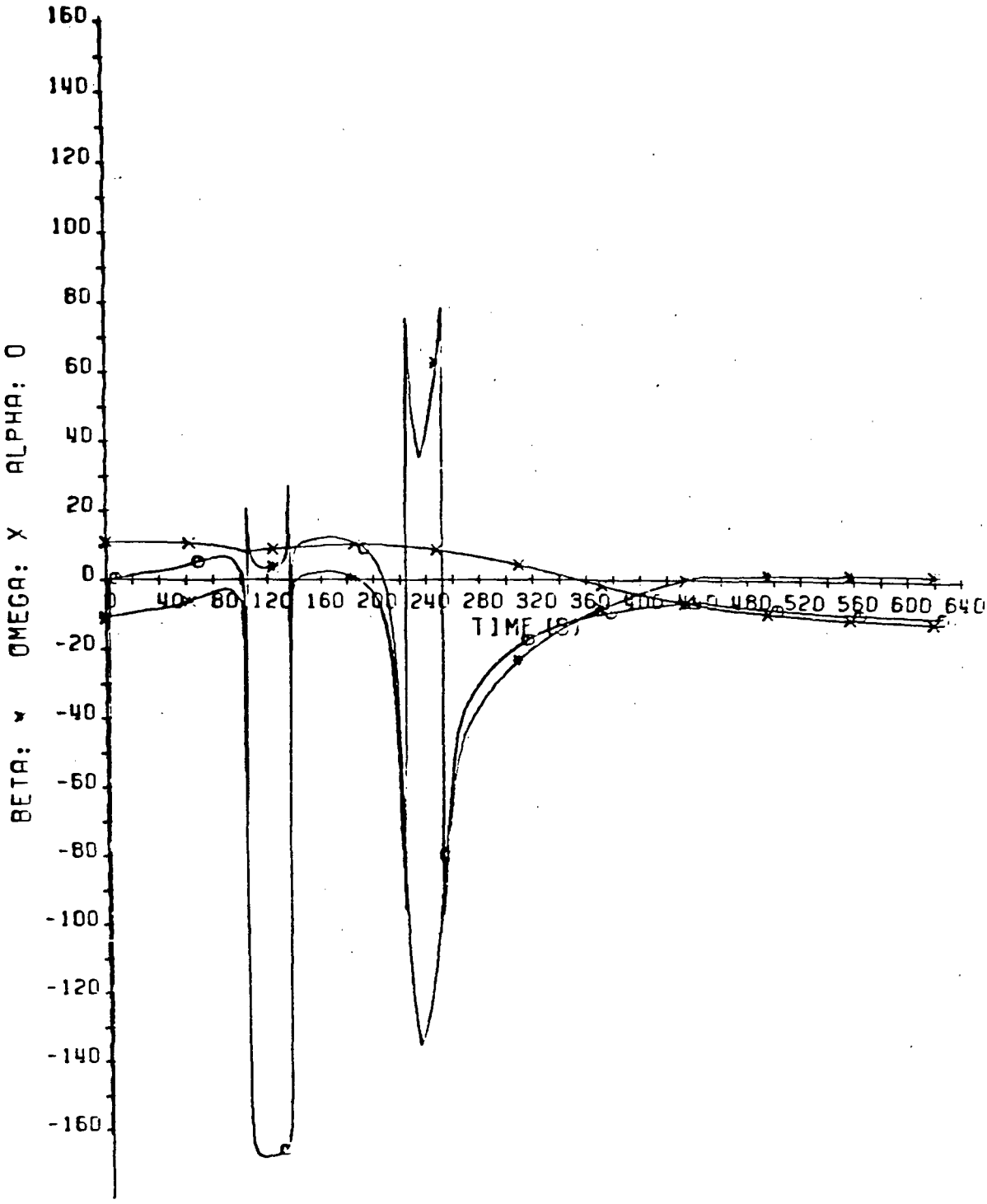


Figure 4.9
Option Two
Navigation Angles Vs. Time: 39 n.m. Perigee Run

4.4.2 Perigee Dispersion Cases With A Thick Atmosphere

The nominal density in these simulation runs is multiplied by a constant factor to increase the aerodynamic forces generated during the atmosphere flight trajectory. Several simulation runs are made with a 125% atmosphere and a 110% atmosphere. A 125% atmosphere means the actual density is constantly 25% greater than the nominal density. Similarly, a 110% atmosphere means the actual density is constantly 10% greater than the nominal atmosphere.

The hinge line lateral guidance algorithm (option two) has better performance than the plane error lateral guidance algorithm (option one) in all the cases with a thick atmosphere. The results of the simulation runs are presented in Table 4.2 and Table 4.3. In all the cases, the total burn magnitude of option two is smaller than both the total burn magnitudes of option one.

The largest ΔV_{min} is 21.21 ft/sec which represents the greatest reduction in the total burn magnitude optained by using option two instead of option one. The largest ΔV_{max} is 22.41 ft/sec which represents the greatest reduction in the total burn magnitude obtained by using option two instead of option one, if the on-board targeting algorithm used by option one does not use an iteration process to find the minimum total burn magnitude. All seven cases with a 125% atmosphere

Table 4.2
Simulation Results For The Perigee Dispersion Cases
With A 125% Atmosphere

Case	Perigee (n.m)	Option One			Option Two			ΔV_{max} (ft/s)	ΔV_{min} (ft/s)
		β (degs)	Plane error (degs)	Roll RV.	β (degs)	Plane error (degs)	Roll RV.		
1	44	87.84	.0430	7	.790	.0190	11	18.54	18.53
2	43	78.81	.0494	7	-10.12	.0102	8	21.60	21.21
3	42	59.89	.0468	6	.404	.00406	8	20.80	18.20
4	41	102.46	.0336	9	18.10	.0273	9	11.47	11.14
5	40	111.71	.0314	9	-.708	.0250	10	14.17	13.23
6	39	96.46	.0461	8	-.235	.0234	10	19.43	19.31
7	37	102.40	.0367	5	13.22	.0191	5	14.38	14.02

have a ΔV_{min} greater than 11 ft/sec, while four of the seven cases with a 110% atmosphere have a ΔV_{min} greater than 11 ft/sec. The final plane error of option two is smaller than the final plane error of option one in all but two cases (9 and 13). The smaller plane error is totally responsible for the reduction of the total burn magnitude obtained by using option two instead of option one in only 3 cases (2, 3, and 10).

Table 4.3
Simulation Results For The Perigee Dispersion Case
With A 110% Atmosphere

Case	Perigee (n.m)	Option One			Option Two			ΔV_{max} (ft/s)	ΔV_{min} (ft/s)
		β (degs)	Plane error (degs)	Roll RV.	β (degs)	Plane error (degs)	Roll RV.		
8	42	97.98	.0151	7	-20.01	.00780	10	5.60	5.53
9	41.0	-37.53	.00575	10	7.81	.0376	11	8.00	4.17
10	40.5	54.27	.0491	6	-13.17	.00034	9	22.41	18.65
11	40.0	89.49	.0422	9	-1.231	.0275	13	17.63	17.63
12	39.5	29.55	.0500	8	-.9151	.0262	11	21.87	11.42
13	39	-44.78	.02395	8	-1.090	.0322	11	9.43	6.41
14	37	102.90	.0320	7	.773	.0209	9	14.46	14.13

Only in two cases did option two require more than three roll reversals than option one.

The thick atmospheres increase the rate of change of the apsidal line position and the total distance over which it moves. Despite this increase in the apsidal line motion, more than half the cases are able

to keep the magnitude of β less than 2 degrees. Case 5 is typical of the cases which are able to keep the magnitude of β less than 2 degrees. The commanded roll angle history and the navigation angles histories of option two for case 5 are shown in Figure 4.10 and Figure 4.11. Four roll reversals are commanded by option two to control the hinge line position. Two roll reversals are commanded to keep the hinge line outside the exclusion zone, while the last two are commanded to keep β inside the deadband.

The final β of option two is positive and outside the deadband limits in three cases (4, 7, and 9). The inability to drive β to zero in these cases is caused by β having a positive value just before the apsidal line position starts to move rapidly. Since β is positive, the apsidal line is moving rapidly away from the hinge line. The out-of-plane lift forces available are not large enough for the hinge line to catch up to the apsidal line; therefore, the final β is outside the deadband limits and is positive. This problem can be corrected by preventing β from attaining a positive value towards the end of the eta control phase; however, this can also increase the total number of roll reversals required by option two which is undesirable.

The commanded roll angle history and the navigation angles histories of option two for case 7 are shown in Figure 4.12 and Figure 4.13. Two roll reversals are commanded by option two to control the hinge line

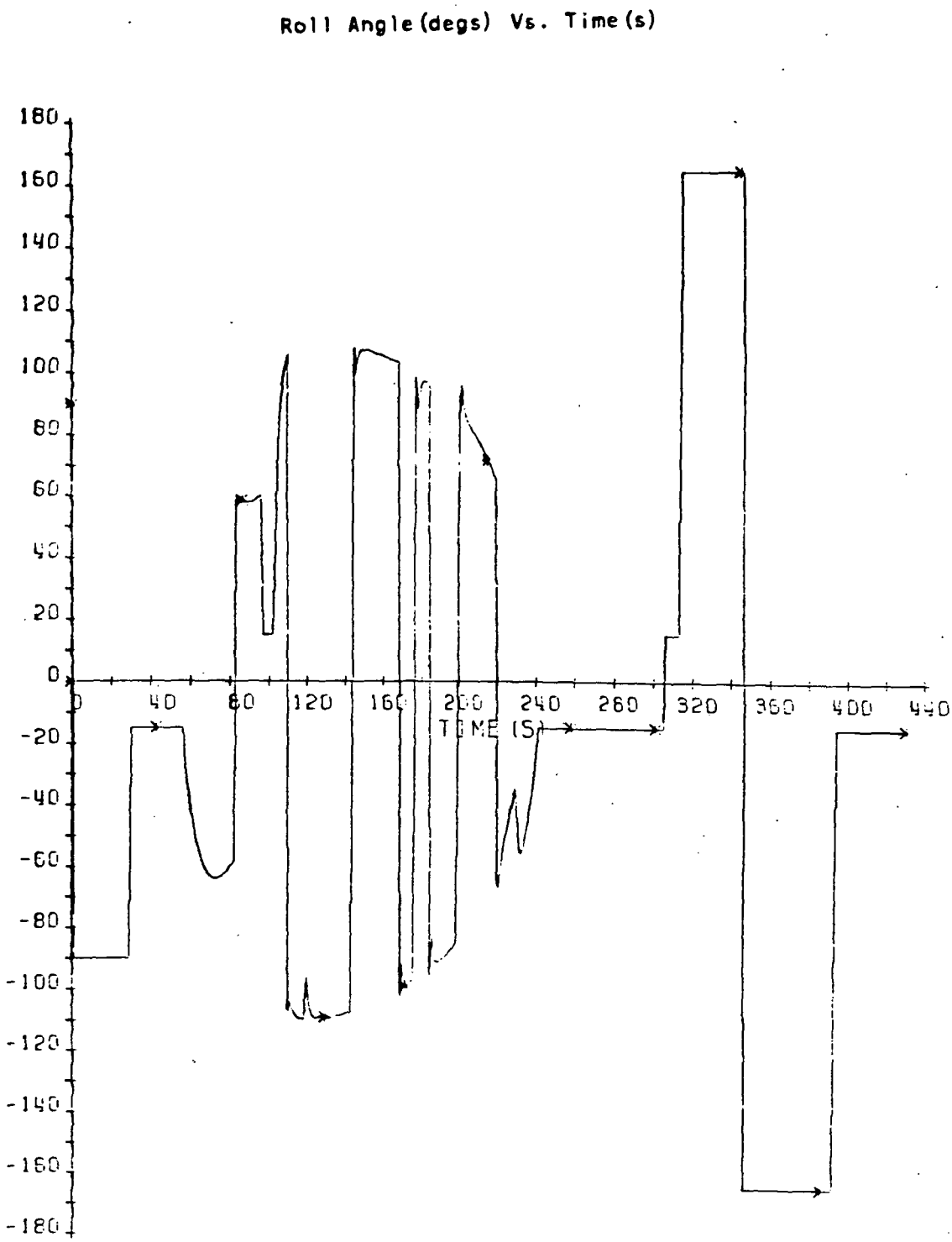


Figure 4.10
Option Two
Roll Angle Vs. Time: 40 n.m. Perigee with 125% Atmosphere

Navigation Angles (degs) Vs. Time (s)

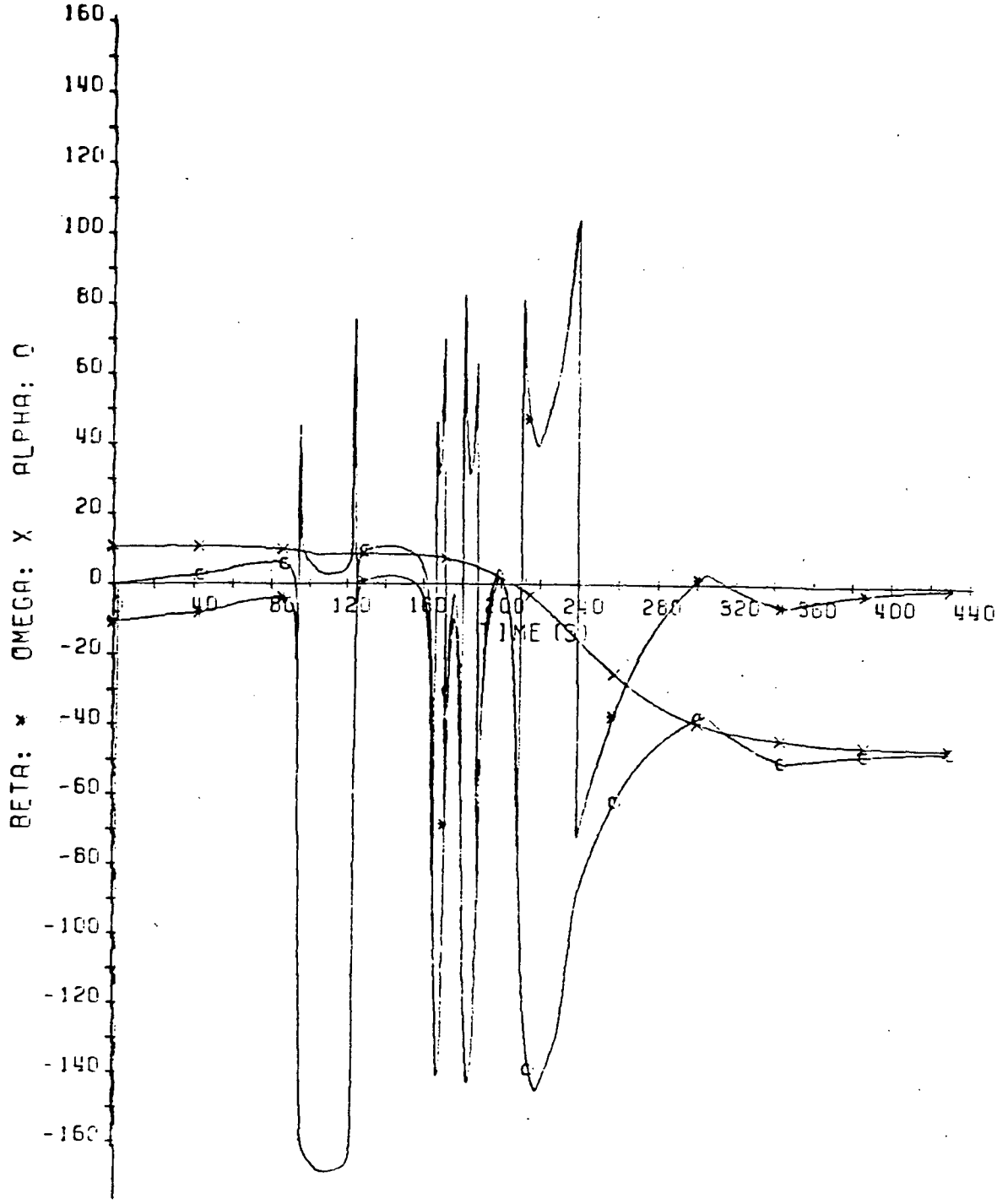


Figure 4.11
Option Two
Navigation Angles Vs. Time: 40 n.m. Perigee with 125% Atmosphere

position. Both these roll reversals are commanded to keep the hinge line outside the exclusion zone. No roll reversals are needed to keep β inside the deadband. Despite the failure to keep β inside the deadband, the total burn magnitude of option two is smaller than both total burn magnitudes of option one.

The beta control phase did not start in time to drive β to zero before the OTV left the atmosphere in case 2. As a result, the final value of β in option two is -10.12 degrees. This problem can be corrected by enlarging the exclusion zone. Unfortunately, the total number of roll reversals will be increased by enlarging the exclusion zone which is undesirable. The reduction in the total burn magnitude obtained by using option two instead of option one is mainly due to the smaller plane error of option two, so the large magnitude of β did not significantly affect the performance of the on-board targeting algorithm. There is no performance penalty in obtaining the smaller plane error of option two, since option two only requires one more roll reversal than option one to obtain this smaller plane error.

The commanded roll angle history and the navigation angles histories of option two for case 2 are given in Figure 4.14 and Figure 4.15. Four roll reversals are commanded to control the hinge line position. Three roll reversals are commanded to keep the hinge line outside the exclu-

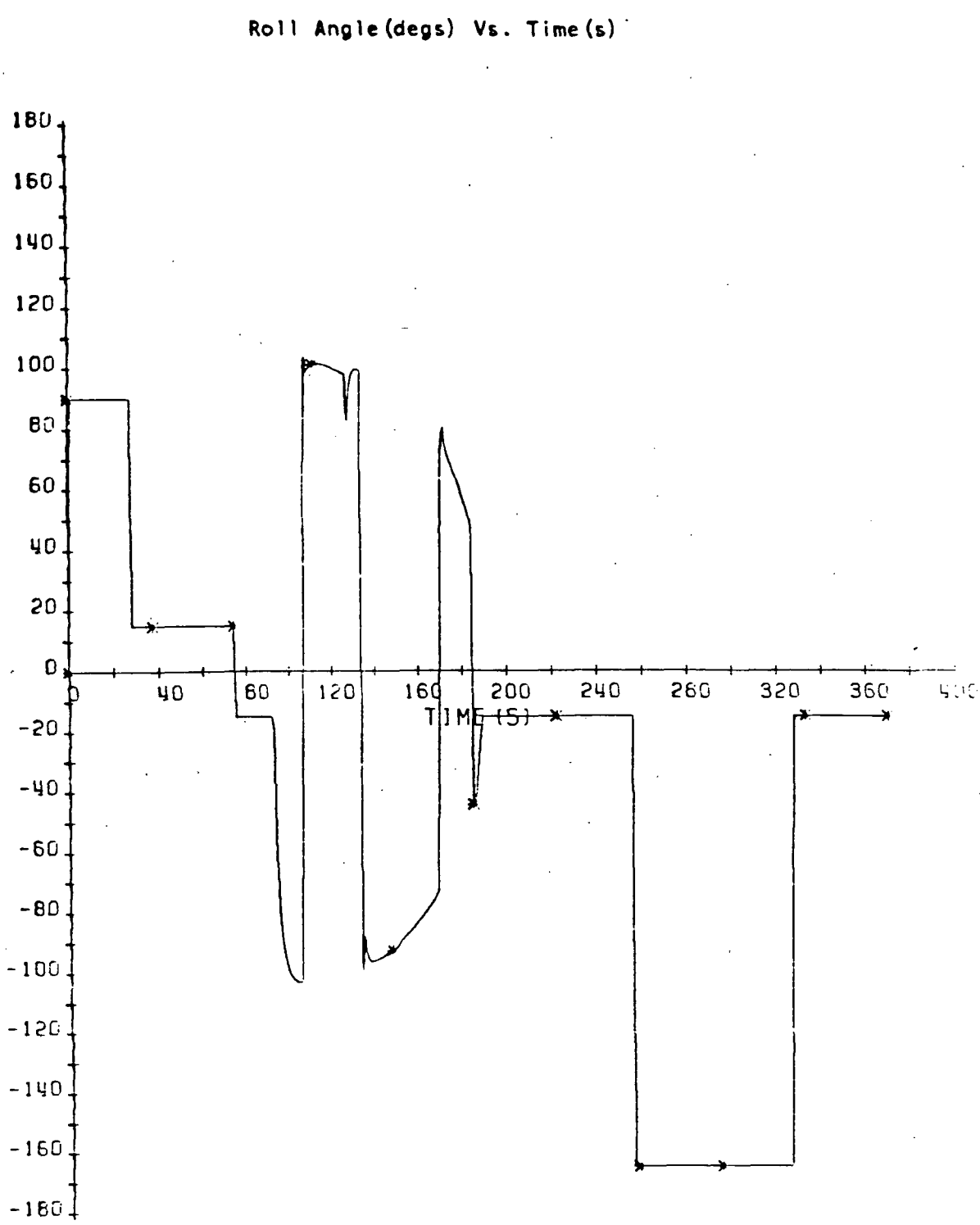


Figure 4.12
Option Two
Roll Angle Vs. Time: 37 n.m. Perigee with 125% Atmosphere

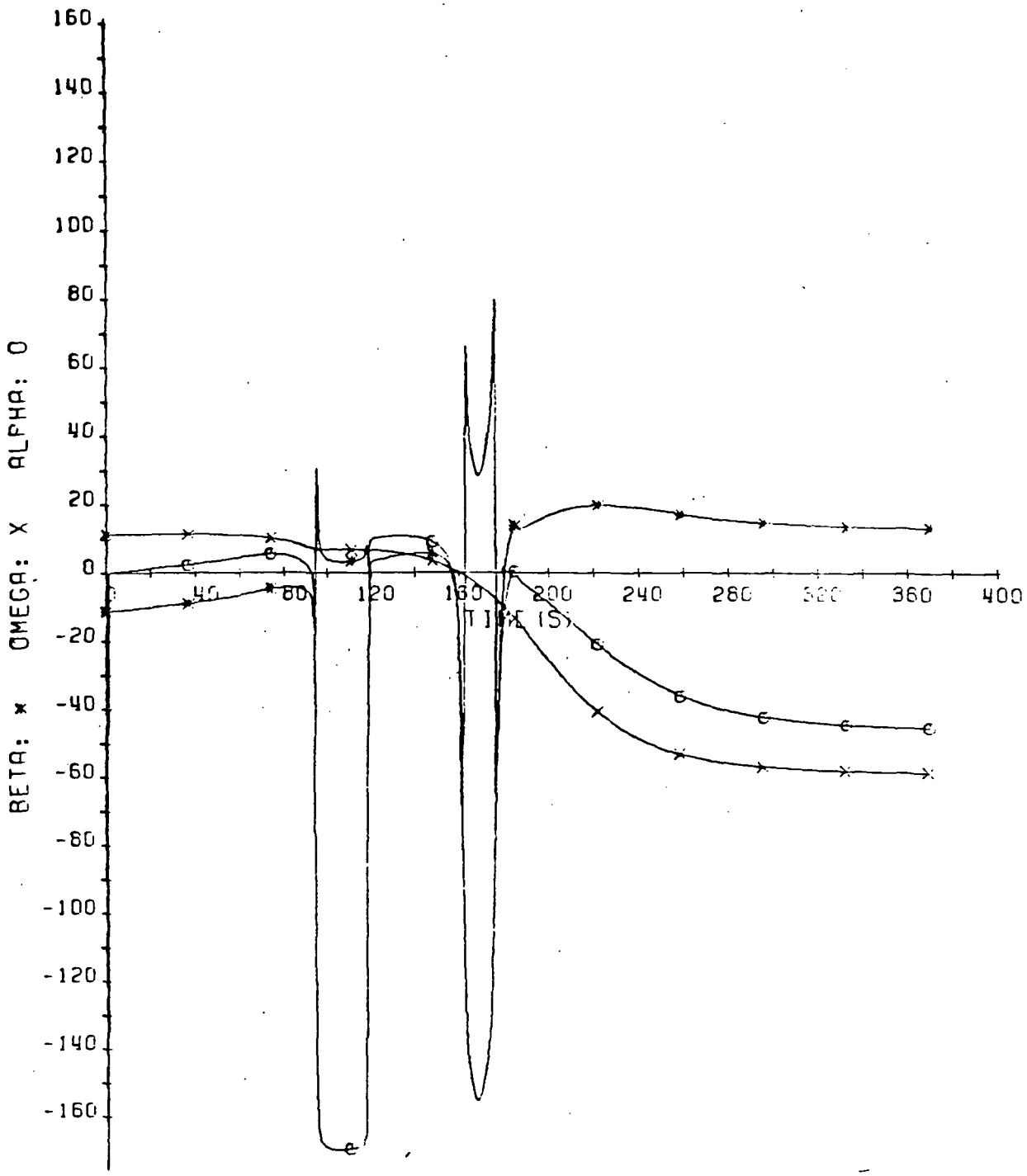


Figure 4.13
Option Two
Navigation Angles Vs. Time: 37 n.m. Perigee with 125% Atmosphere

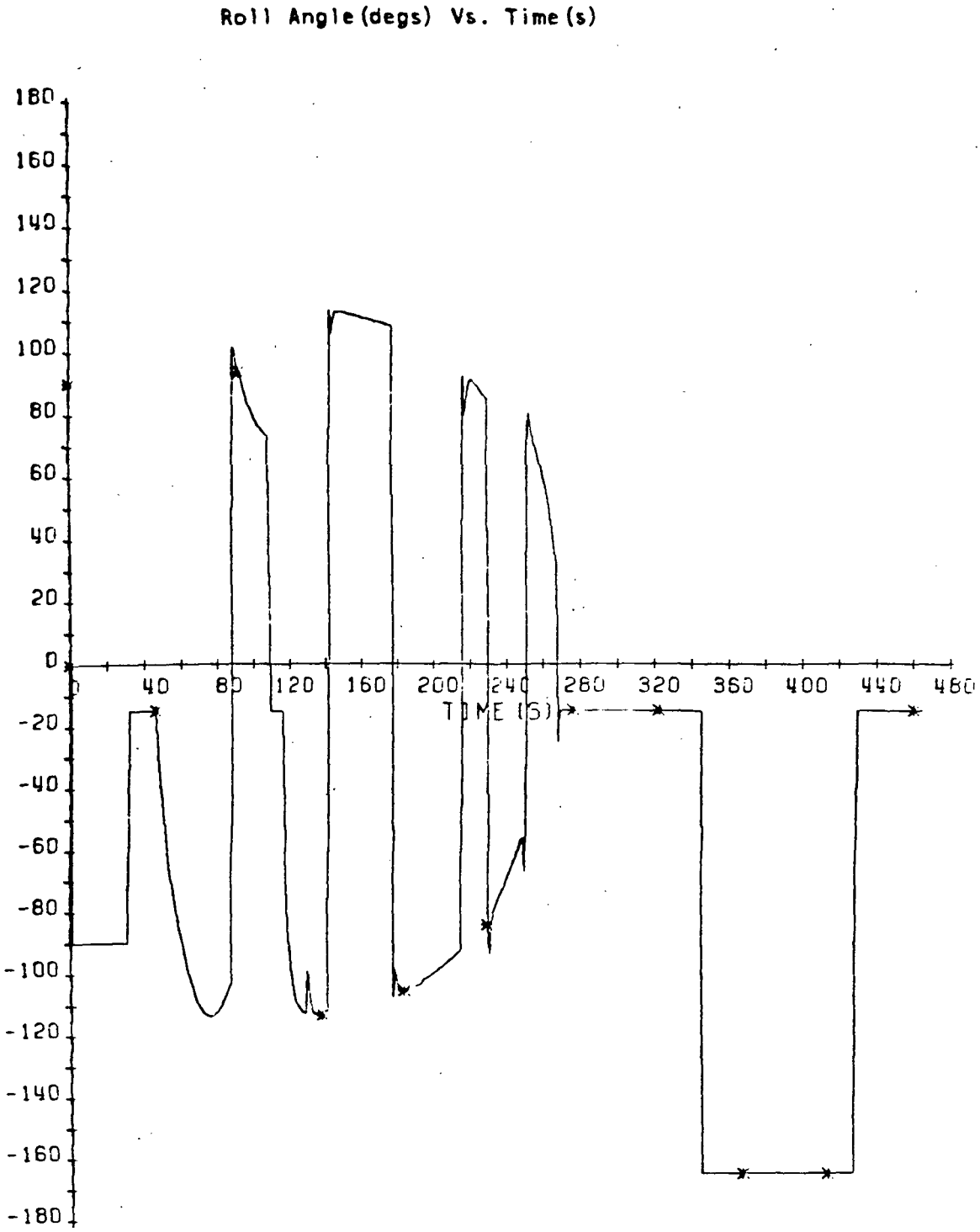


Figure 4.14
Option Two
Roll Angle Vs. Time: 43 n.m. Perigee with 125% Atmosphere

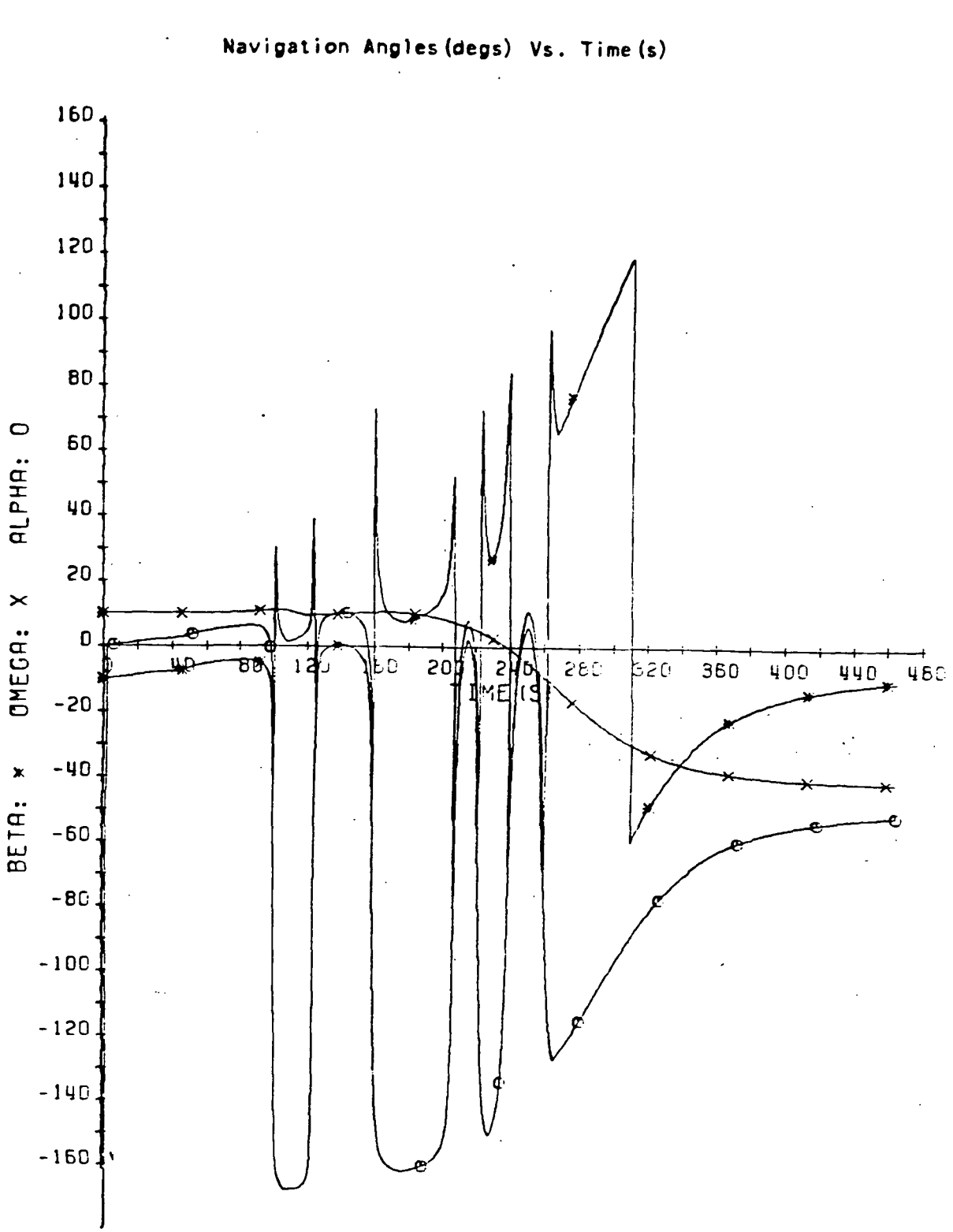


Figure 4.15
Option Two
Navigation Angles Vs. Time: 43 n.m. Perigee with 125% Atmosphere

sion zone, and one roll reversal is commanded to keep β inside the deadband.

The small plane error control is used in cases 3, 8, and 10. The desired plane error needed for β to equal zero is less than 0.01 degrees in these cases, because the position out-of-plane error is small at the start of the eta control phase. Since the plane error is less than 0.01 degrees, the magnitude of β is not as critical in these cases. The small plane error is totally responsible for the large reduction in the total burn magnitude obtained by using option two instead of option one in cases 3 and 10. Option two requires just two more roll reversals for case 3 and three more roll reversals for case 10 than option one to obtain the smaller plane error. In case 8, the small plane error and the proximity of the hinge line to the apsidal line are equally responsible for the reduction of the total burn magnitude obtained by using option two instead of option one. The large magnitude of β did not adversely affect the on-board targeting algorithm in these three cases, since the plane error is small.

The desired plane error falls below 0.01 degrees but stays above 0.001 degrees during the beta control phase in case 8. As a result, the original beta control phase is no longer used, but instead the eta control phase is used. When the rate of change of the longitude of the hinge line is less than 1.5 degrees/sec, then the beta control phase is

used again and a roll reversal is commanded if β is outside the deadband and the flag is not set. This modified version of the beta control phase effectively enlarges the deadband limits. No change is made to the eta control phase, since the desired plane error stays above 0.01 degrees during the eta control phase.

The commanded roll angle history and the navigation angles histories of option two for case 8 are shown in Figure 4.16 and Figure 4.17. Six roll reversals are commanded by option two to control the hinge line position. Four roll reversals are commanded to keep the hinge line outside the exclusion zone. The last two roll reversals are commanded by the modified version of the beta control phase.

The small plane error logic is used during the eta and beta control phases in case 3. The desired plane error falls below 0.01 degrees during the eta control phase. As a result, the eta control phase is no longer used, but instead the modified plane error control phase described in Section 3.5 is used. The modified version of the beta control phase is used in case 3 as in case 8, since the desired plane error falls beneath 0.01 degrees but stays above 0.001 degrees during the beta control phase.

The commanded roll angle history and the navigation angles histories of option two for case 3 are shown in Figure 4.18 and Figure 4.19. Four

Roll Angle(degs) Vs. Time(s)

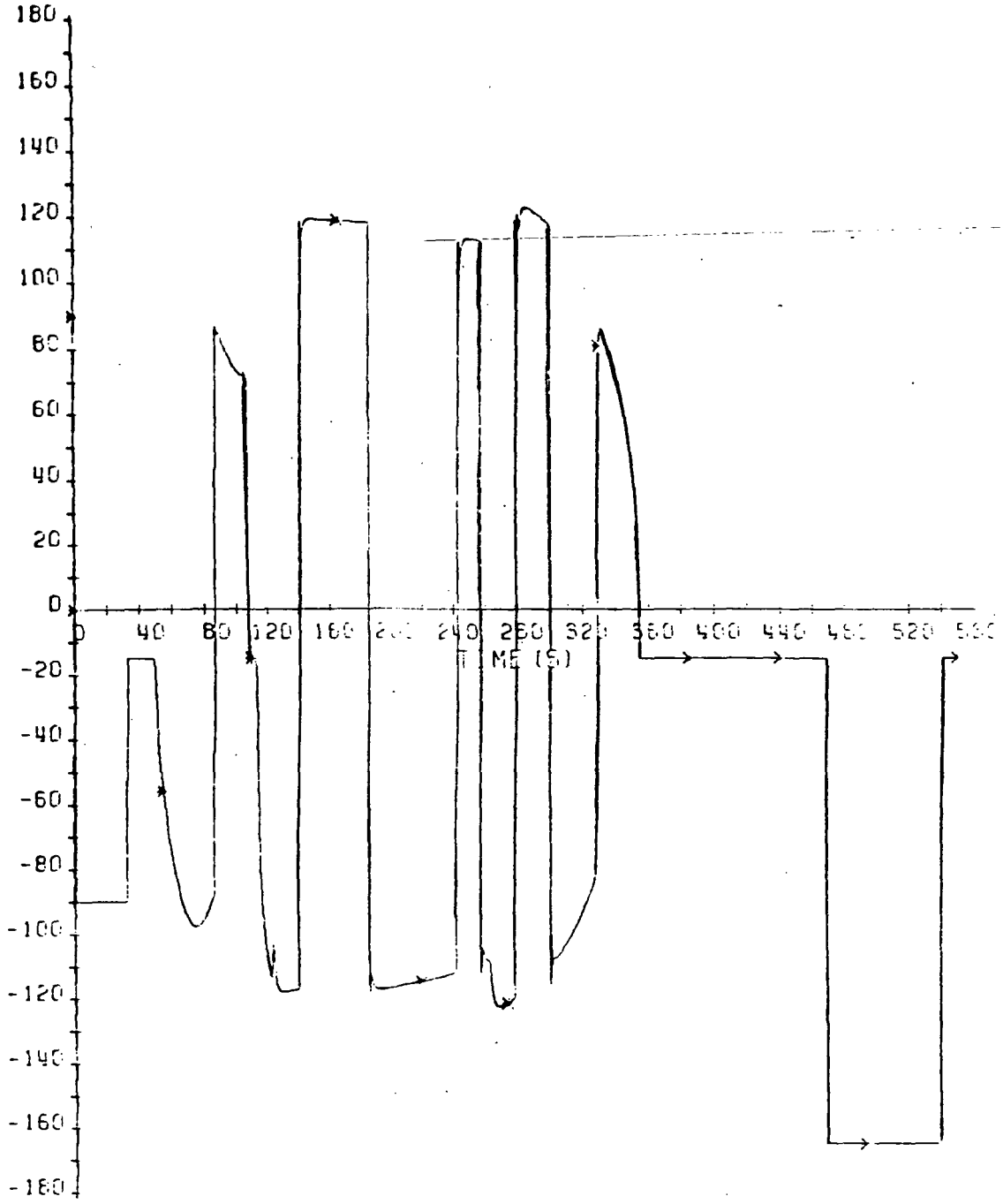


Figure 4.16
Option Two
Roll Angle Vs. Time: 42 n.m. Perigee with 110% Atmosphere

Navigation Angles (degs) Vs. Time (s)

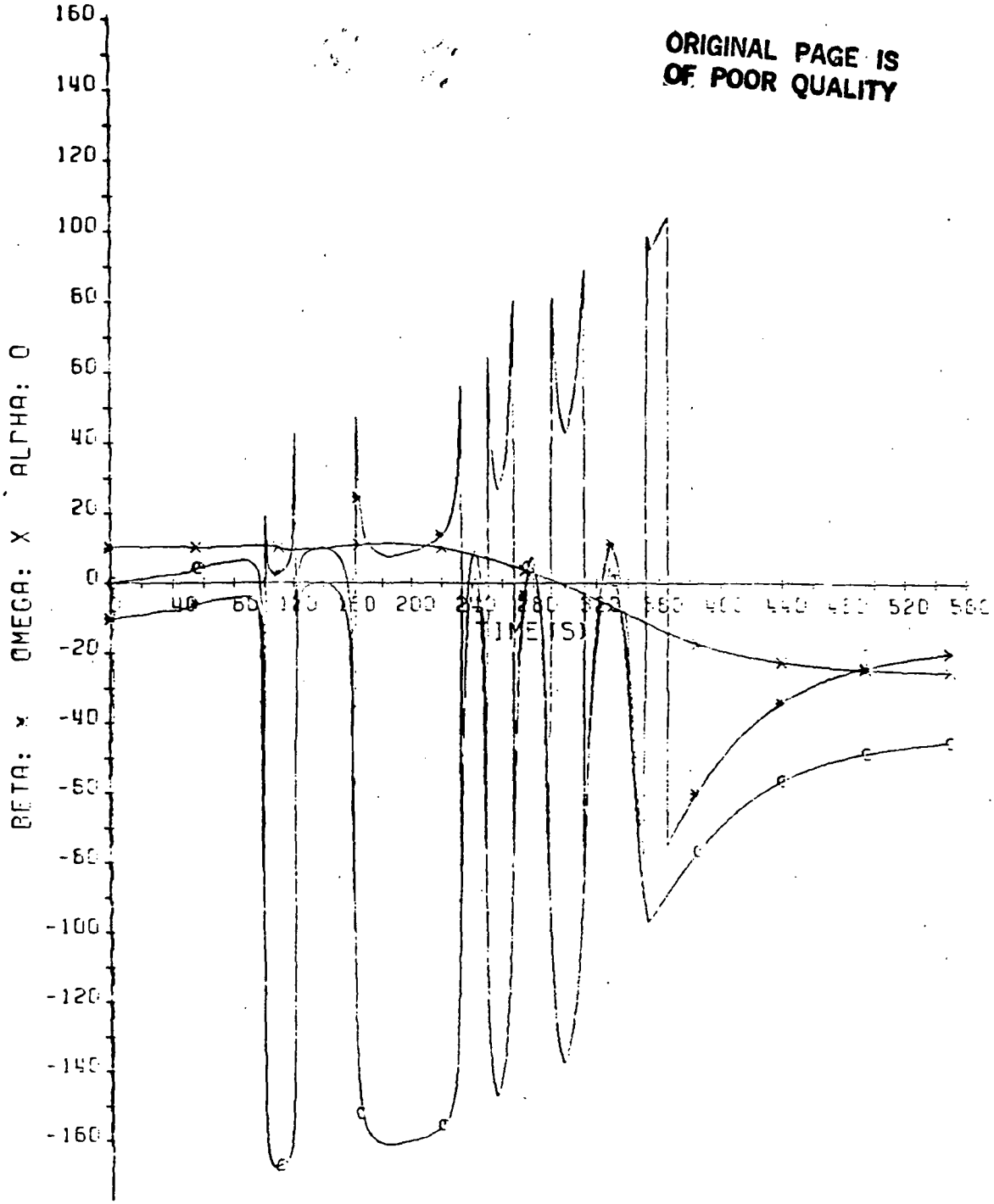


Figure 4.17
Option Two
Navigation Angles Vs. Time: 42 n.m. Perigee with 110% Atmosphere

C-2

roll reversals are commanded by option two to control the hinge line position. Two roll reversals are commanded by the modified version of the plane error control phase. The last two roll reversals are commanded by the modified version of the beta control phase.

The small plane error logic is used again during the eta and beta control phases in case 10; however, a different modified version of the beta control phase is used. The desired plane error falls beneath 0.001 degrees during the beta control phase. As a result, the original beta control logic is no longer used, but instead the modified plane error control phase is used. When the rate of change of the longitude of the hinge line is less than 0.2 degrees/sec, the beta control phase is used again, and a roll reversal will be commanded if β is outside the deadband and the flag is not set. This second modified version effectively enlarges the deadband limits and places no restrictions on the hinge line position. The desired plane error during the eta control phase is beneath 0.01 degrees, and the modified version of the plane error logic is used for case 10 as in case 3.

The commanded roll angle history and the navigation angles histories of option two for case 10 are shown in Figure 4.20 and figure 4.21. Figure 4.21 illustrates the rapid rate of change of the hinge line position when the plane error is less than 0.001 degrees. The rapid movement of the hinge line makes confining β to a small deadband impossible

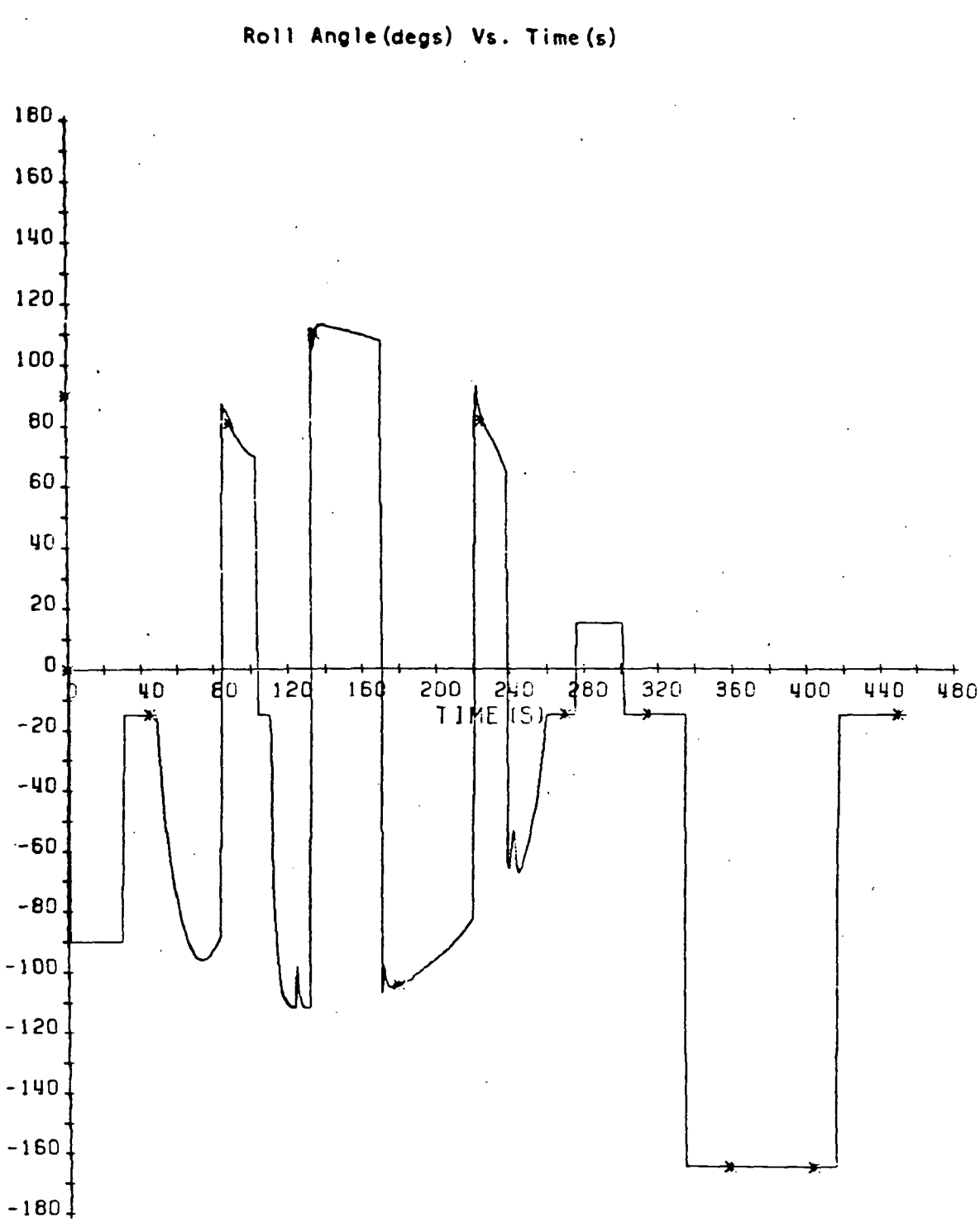


Figure 4.18
Option Two
Roll Angle Vs. Time: 42 n.m. Perigee with 125% Atmosphere

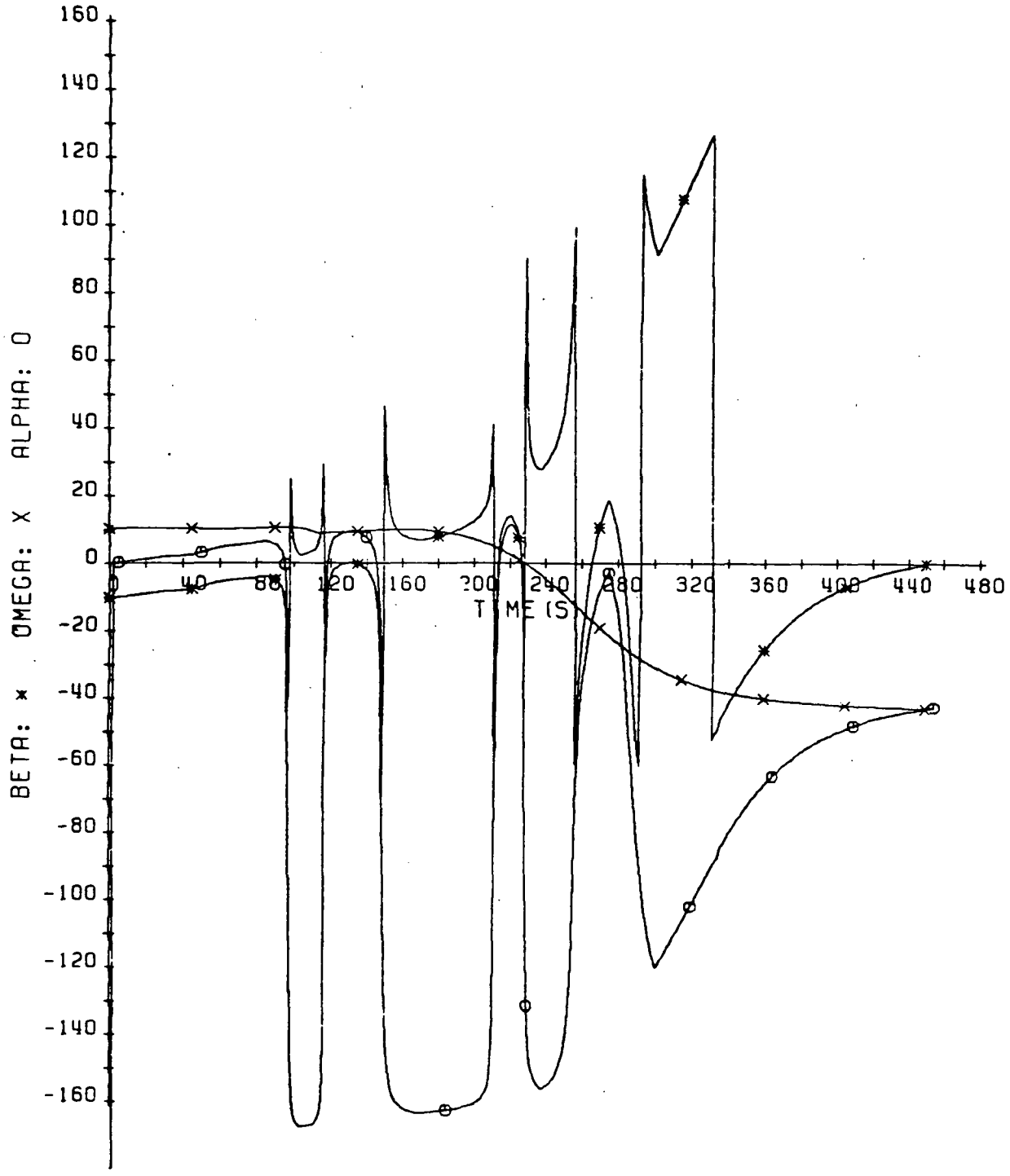


Figure 4.19
Option Two
Navigation Angles Vs. Time: 42 n.m. Perigee with 125% Atmosphere

without using numerous roll reversals. Five roll reversals are commanded by option two to control the position of the hinge line. Two roll reversals are commanded by the modified version of the plane error control phase. The last three roll reversals are commanded by the second modified version of the beta control phase. The last three roll reversals can probably be eliminated without effecting the performance of the on-board targeting algorithm, since the reduction in the total burn magnitude of option two is only 0.1 ft/sec by driving β to zero for case 10.

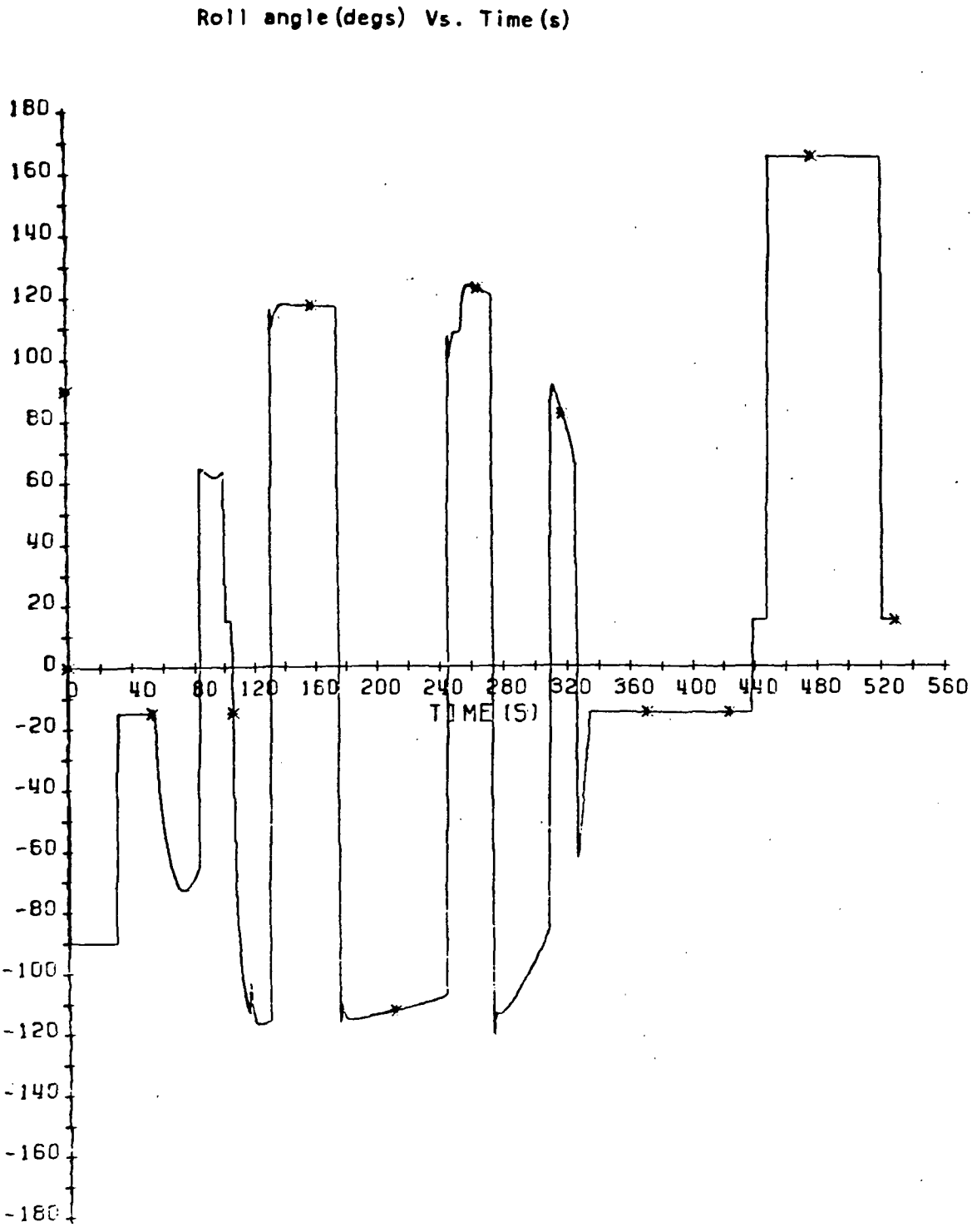


Figure 4.20
Option Two
Roll Angle Vs. Time: 40.5 n.m. Perigee With 110% Atmosphere

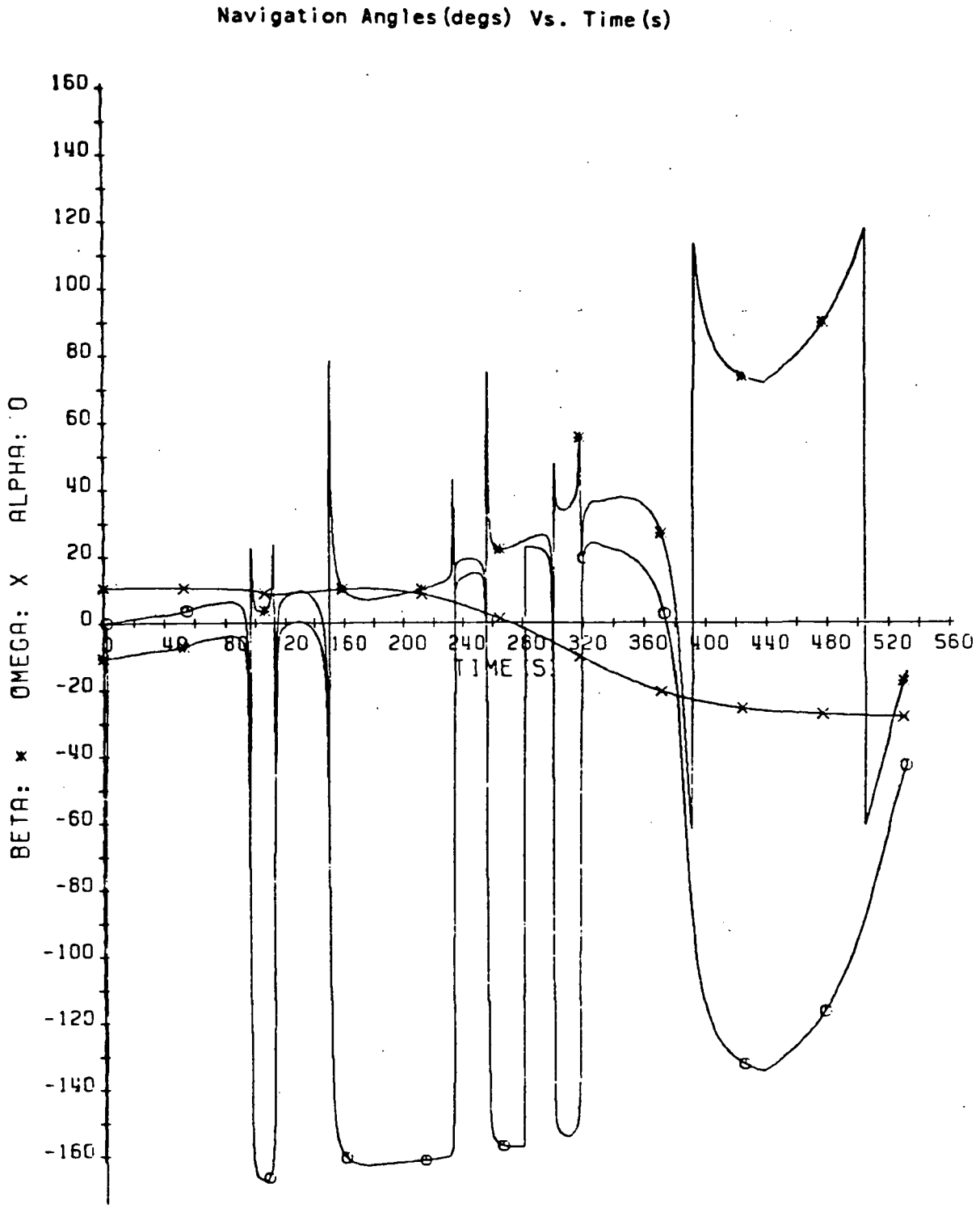


Figure 4.21
Option Two
Navigation Angles Vs. Time: 40.5 n.m. Perigee with 110% Atmosphere

4.4.3 Pot-Hole Cases

The lateral guidance algorithm was evaluated for eight different pot-holes. The actual density is decreased by 15% from the nominal density during the pot-hole in each case. The placement of the pot-hole is selected to produce the worst possible performance of the hinge line lateral guidance algorithm. Each pot-hole ends just before the eta control phase starts (ie. VELBIAS2 = 1800 ft/sec), but the starting point (ie. VELBIAS1) is varied for each pot-hole. By increasing VELBIAS1, the position out-of-plane error at the start of the eta control phase is increased which produces a larger desired plane error.

The performance of the hinge line lateral guidance algorithm (option two) is only marginally better than the performance of the plane error lateral guidance algorithm (option one) in the pot-hole cases. The results of the simulation runs are presented in Table 4.4. Despite the larger plane error of option two, the total burn magnitude of option two is smaller than the maximum total burn magnitude of option one in seven cases. Unfortunately, the total burn magnitude of option two is slightly larger than the minimum total burn magnitude of option one in five cases.

The largest ΔV_{min} is 3.57 ft/sec which represents the greatest reduction in the total burn magnitude by using option two instead of

option one. The largest increase in the total burn magnitude by using option two instead of option one is 6.40 ft/sec. The largest ΔV_{max} is 22.89 ft/sec which represents the greatest reduction in the total burn magnitude by using option two instead of option one, if the on-board targeting algorithm of option one does not use an iteration process to find the minimum total burn magnitude. The largest increase in the total burn magnitude of option two with respect to the maximum total

Table 4.4
Simulation Results For The Pot-Hole Cases

Pot-hole	VELBS1 (ft/s)	Option One			Option Two			ΔV_{max} (ft/s)	ΔV_{min} (ft/s)
		β (degs)	Plane error (degs)	Roll RV.	β (degs)	Plane error (degs)	Roll RV.		
1	2400	72.38	.00648	6	.0428	.0250	9	2.07	1.94
2	2800	15.43	.0339	6	1.089	.0333	7	14.07	3.57
3	3200	9.36	.0572	7	3.92	.0494	7	22.89	3.15
4	3400	-2.74	.0586	7	5.05	.0617	7	17.96	-2.20
5	3600	-3.87	.0537	7	4.45	.0638	7	19.25	-1.47
6	3800	-31.50	.0457	7	7.45	.121	7	4.79	-4.15
7	4200	-36.49	.0483	7	7.07	.137	7	3.14	-4.81
8	4600	-49.84	.0470	7	7.03	.158	7	-1.94	-6.40

burn magnitude of option one is 1.94 ft/sec. The plane error of option two is greater than the plane error of option one in six cases. In only two cases does option two require more roll reversals than option one.

The final plane error of option two is roughly equal to the final plane error of option one in cases 2 and 3. The total burn magnitude of option two is significantly less than the maximum total burn magnitude of option one for both cases. Since the final plane errors of option one and option two are roughly equal, the large value for ΔV_{max} is totally attributed to the small magnitude of β in option two. The final magnitude of β in option one also happens to be small is cases 2 and 3. As a result, the total burn magnitude of option two is only slightly less than the minimum total burn magnitude of option one.

The final plane error of option two is significantly larger than the final plane error of option in four cases (1,6,7, and 8). The total burn magnitude of option two is less than both total burn magnitudes of option one in case 1 only. The smaller final β of option two in case 1 is responsible for the reduction in the total burn magnitude of option two. The total burn magnitude of option two is larger than the minimum total burn magnitude of option one for the other three cases (6,7,and 8) and the maximum total burn magnitude of option one just for case 8. The inability to drive β to zero in the three cases is responsible for the poor performance of the on-board targeting algorithm. Even though the

final β is only six degrees outside the deadband, the total burn magnitude of option two can still be significantly reduced if β is smaller. If β is inside the deadband for case 8, the total burn magnitude of option two can be reduced by at least 8.5 ft/sec which will make it less than both total burn magnitudes of option one. Similarly for cases 6 and 7, if β is inside the deadband, the total burn magnitude of option two will be less than both total burn magnitudes of option one.

The final plane error of option two is slightly larger than the final plane error of option one for two cases (4 and 5). The total burn magnitude of option two is significantly less than the maximum total burn magnitude of option one for both cases. The final β of option one is less than the final β of option two in these cases. As a result, the total burn magnitude of option two is greater than the minimum total burn magnitude of option one for both cases.

4.4.4 Pot-Hole Cases With A Thick Atmosphere

The lateral guidance algorithm is evaluated for pot-holes in various thick atmospheres. Thick atmospheres are added to the pot-holes of cases 2,4, and 5 from Section 4.4.3. The actual density is still decreased by 15% from the nominal density when in the pot-hole, but the actual density is increased by a constant factor for the flight trajectory outside the pot-hole (see Figure 4.22). A 110% atmosphere means that the actual density is 10% greater than the nominal density for the flight trajectory outside the pot-hole. By increasing the actual density outside the pot-holes, the ability to change the hinge line position, the plane error, and the position and velocity out-of-plane errors is increased.

The hinge line lateral guidance algorithm (option two) has better performance than the plane error lateral guidance algorithm (option one) in all the cases with a pot-hole in a thick atmosphere. The results of the simulation runs are presented in Table 4.5, Table 4.6, and Table 4.7. In all the cases, the total burn magnitude of option two is smaller than both total burn magnitudes of option one.

The largest ΔV_{min} is 16.49 ft/sec which represents the greatest reduction in the total burn magnitude obtained by using option two instead of option one. The largest ΔV_{max} is 24.64 ft/sec which repres-

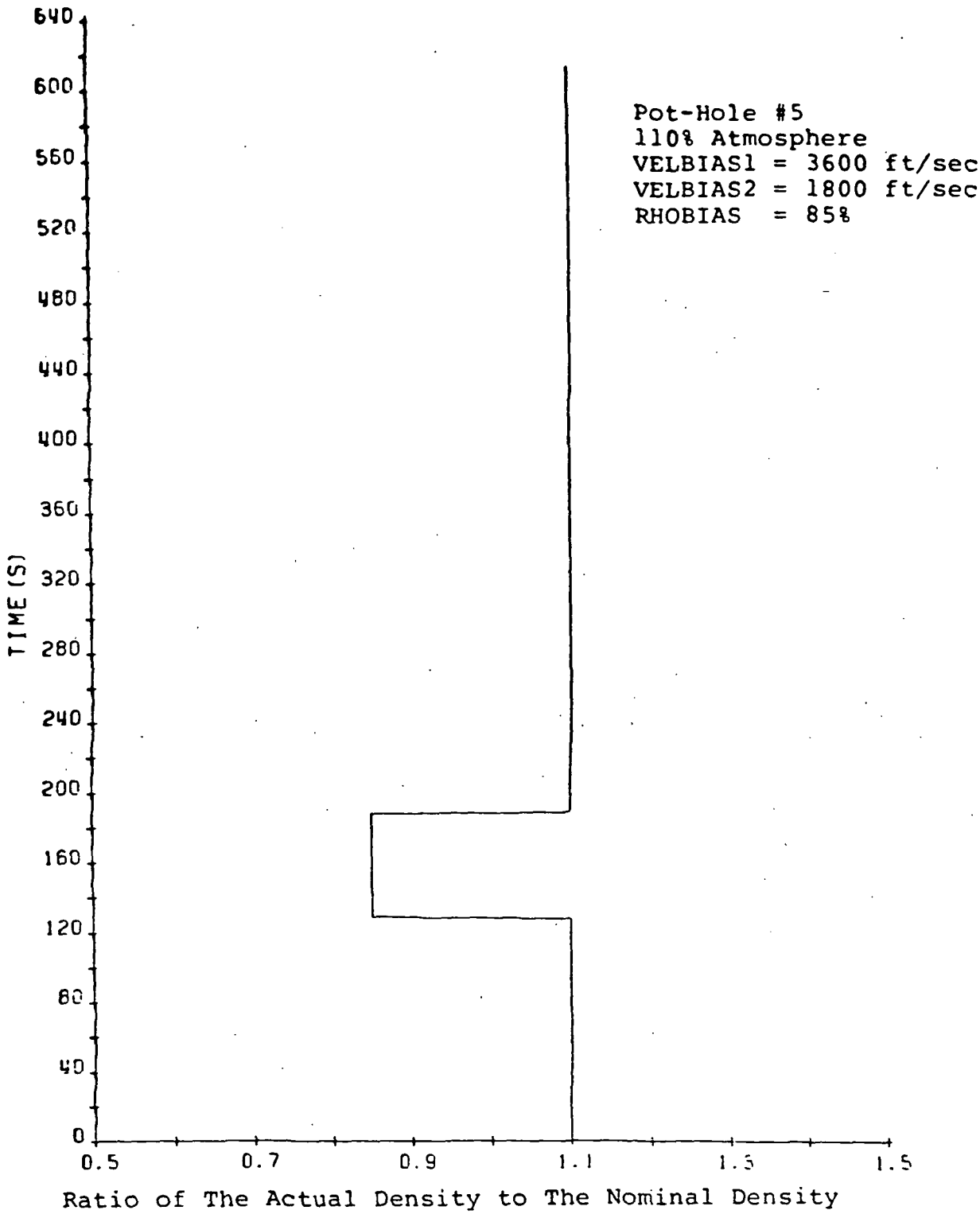


Figure 4.22
Typical Density Profile With A Pot-Hole Present In A Thick Atmosphere

ents the greatest reduction in the total burn magnitude obtained by using option two instead of option one, if the on-board targeting algorithm used by option one does not use an iteration process to find the minimum total burn magnitude. The final plane error of option two is larger than the final plane error of option one in half of the cases. Despite the larger plane error of option two, the total burn magnitude of option two is smaller than both burn magnitudes of option one. This reduction in the total burn magnitude is a result of the smaller β of option two. In case 1, the plane error of option two is almost twice the size of the plane error of option one, but there is still a large

Table 4.5
Simulation Results For Pot-Hole 2 With Different Thick Atmospheres

Case	Option One			Option Two			ΔV_{max}	ΔV_{min}
	β (degs)	Plane error (degs)	Roll RV.	β (degs)	Plane error (degs)	Roll RV.	(ft/s)	(ft/s)
1 105% atmos.	-65.64	.0351	8	2.23	0.0619	9	12.64	11.35
2 110% atmos.	18.43	.0686	9	8.21	0.0610	9	24.64	5.72
3 120% atmos.	22.14	.0280	9	.149	0.0170	13	11.93	4.49
4 125% atmos.	22.07	.0551	9	4.43	0.0380	9	21.76	7.55
5 130% atmos.	21.156	.0287	9	.560	0.0203	9	12.33	4.49

reduction of the total burn magnitude by using option two (ΔV_{max} = 12.64 ft/sec and ΔV_{min} = 11.35 ft/sec). Option two requires more roll reversals than option one in four cases; however, option two requires fewer roll reversals than option one in five cases.

Table 4.6
Simulation Results For Pot-Hole 4 With Different Thick Atmospheres

Case	Option One			Option Two			ΔV_{max}	ΔV_{min}
	β (degs)	Plane error (degs)	Roll RV.	β (degs)	Plane error (degs)	Roll RV.	(ft/s)	(ft/s)
6 110% atmos.	-55.26	.0481	7	9.76	.0771	7	13.93	10.45
7 120% atmos.	97.04	.0387	9	1.63	.0338	11	16.61	16.49
8 125% atmos.	102.26	.0566	8	20.70	.0693	7	13.02	12.50
9 130% atmos.	105.15	.0369	10	.351	.0338	9	16.61	16.08

Table 4.7
Simulation Results For Pot-Hole 5 With Different Thick Atmospheres

Case	Option One			Option Two			ΔV_{max}	ΔV_{min}
	β (degs)	Plane error (degs)	Roll RV.	β (degs)	Plane error (degs)	Roll RV.	(ft/s)	(ft/s)
10 110% atmos.	-29.29	.0507	7	8.76	.0794	7	15.19	4.57
11 115% atmos.	110.08	.0505	8	13.58	.0787	7	12.71	11.48
12 120% atmos.	-46.54	.0282	10	11.06	.0468	11	8.39	5.12
13 125% atmos.	-28.80	.0445	7	4.94	.0529	11	16.31	6.71
14 130% atmos.	97.81	.0493	10	15.65	.0481	9	15.87	15.68

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH

A lateral guidance algorithm based on controlling the hinge line position has been developed and tested in this thesis. The on-board targeting algorithm associated with the hinge line lateral guidance algorithm is concise and requires less computing time than the one associated with the plane error lateral guidance algorithm. Equations have been developed which describe the varying nature of the hinge line and determine the hinge line position. Simple relationships between the plane error, the desired hinge line position, the position out-of-plane error, and the velocity out-of-plane error were found.

The hinge line lateral guidance algorithm (option two) had better performance than the plane error lateral guidance algorithm (option one) over a wide range of operating conditions. Despite the larger final plane error of option two in some cases, the total burn magnitude was reduced by using option two instead of option one in almost every case.

There was no performance penalty for using option two instead of option one, since the total number of roll reversals was not significantly increased by using option two.

The total burn magnitude of option two is less than the minimum total burn magnitude of option one for the majority of the operating conditions tested. In the cases where the total burn magnitude of option two was greater than the minimum total burn magnitude of option one, the increases were significantly less than the reductions in the total burn magnitude obtained by using option two in the other cases. Furthermore, the operating conditions which produced the increases in the total burn magnitudes were specifically selected to produce poor performance for option two and have a low probability of occurring in the actual environment.

The on-board targeting algorithm used an iteration process to find the minimum total burn magnitude of option one. If the size of the on-board flight computer is too small, the minimum total burn magnitude of option one could not be found. Under these circumstances, the on-board targeting algorithm which produced the maximum total burn magnitude would be used by option one. The total burn magnitude of option two was less than the maximum total burn magnitude of option one for all the cases tested except one. In that one case, the increase in the total burn magnitude by using option two instead of option one was insignif-

icant. The reduction in the total burn magnitude by using option two instead of option one was greater when the on-board targeting algorithm of option one could only find the maximum total burn magnitude.

The hinge line lateral guidance algorithm was able to keep β in the deadband for most of the cases tested. The inability to keep β inside the deadband was responsible for the few cases where the total burn magnitude of option two was greater than the total burn magnitude of option one. The reason for the inability to keep β inside the deadband was similar for most of the cases. Eta was positive just before the apsidal line position started to change rapidly. As a result, the apsidal line was rapidly moving away from the hinge line. Unfortunately, the out-of-plane lift forces present were insufficient to drive the hinge line position to the apsidal line position. Consequently, the final β was outside the deadband for these cases.

The large final β presents a problem which must be corrected to obtain greater reductions in the total burn magnitude. One way to correct this problem is to decrease the upper limits of the deadband to take into account the apsidal line movement. Altering the criteria for when to switch to the eta control phase from the plane error control phase is another way to prevent the final β from being outside the deadband. The criteria should be altered to take into account the position out-of-plane error magnitude. By decreasing the position out-of-plane

error magnitude when the eta control phase starts, the desired plane error will also be smaller. The smaller plane error will decrease the actual plane error which increases the ability to change the hinge line position. Both these methods need to be investigated to see if they will improve the performance of the hinge line lateral guidance algorithm.

A totally different approach in designing a hinge line lateral guidance algorithm might result in greater reductions in the total burn magnitude and fewer required roll reversals. If the hinge line is driven to the predicted final apsidal line position instead of the current apsidal line position, the inability to keep β inside the deadband might be eliminated. A variational equation must be developed to predict the final apsidal line position given the current conditions and the expected time of flight left in the atmosphere. Unfortunately, the behavior of the apsidal line position is extremely non-linear which makes predicting its final position difficult. However, if a predictor/corrector aerobraking guidance law is being used as in reference [11], the final apsidal line position can be easily obtained. Another alternative in designing a hinge line lateral guidance algorithm is possible if the aerobraking guidance law of reference [11] is being used. The hinge line position can be kept near the current OTV position until the lift forces generated over the remaining trajectory will be sufficient to just drive β to zero. By basing the hinge line lateral

guidance algorithm on this approach, the number of required roll reversals could be greatly reduced, though unexpected density variations could create problems. Both these alternatives to designing a lateral guidance algorithm seem promising for further research.

In conclusion, the work presented in this thesis provides a firm foundation from which to implement a hinge line lateral guidance algorithm on an OTV. Further testing needs to be done to demonstrate decisively the advantage of the hinge line lateral guidance algorithm and to determine the best deadband limits. In particular, the performance of the hinge line lateral guidance algorithm in the presence of navigation errors and finite roll rates must be evaluated to prove completely the effectiveness of the algorithm.

APPENDIX A

SIMULATION COMPUTER PROGRAMS

This appendix contains the source code for the major computer programs used in testing the lateral guidance algorithm. Not included is the program GCH.BURNS4A which calculates the post-aerobraking burn magnitudes discussed in Section 2.5. Included in this order are:

GCH.DRIVET7- driver

GCH.SIMT7- environment simulation

GCH.GUID8C- aerobraking guidance law and lateral guidance algorithm

GCH.ORBITS4A- orbital elements and control parameters calculation

The computer programs are written in MAC which is a language developed at the Charles Stark Draper Laboratory.

Following the source codes is a list of the input values for a nominal run.

```
MAC*GCH.DRIVET7
*****************************************************************
SOURCE      : GCH1752.THESIS.MAC(DRIVET7)
AUTHOR      : H.R. MORTH AND G.C. HERMAN
PURPOSE     : PERFORMS ALL INPUT AND INITIALIZATION OPERATIONS
INPUTS      : RUN CONTROL VARIABLES
OUTPUTS     : INITIAL POSITION AND VELOCITY FOR TRUE AND
              NAVIGATION STATES
*****************************************************************

      -
COMMON(CONST)  DUM1,IPOLE,DUM2,MU,RE,J2,DUM3,DUM4,WE,GZERO

COMMON(CABRAKE), ACCEL, WLIM, WDOTLIM, RNAV, VRELNAV, VNAV,
                 INCL, INCLD, LODNAV, LOD, CB, DTSIM, PHI, ROLL,
                 RHO, RHOVAR, PLOTSW, HSTEMP, KRHO,ICNTL, PRTNO,
                 - - -    -
                 R, V, RNAV, VNAV, HPI, TMAX, GUIDRATE,
                                              -
                 NAVSW, LODSW, RDOTNAV, VEX1, IYD, STARTALT,SIZE,
                                                 -
                 NGRAVW, TOUT, FIRSTPASS, ISTART, ACCEL, CBNAV,
                 RHOSTD, LODEST,SWITCH3BS,LIFTSW1,LIFTSW2,
                 LIFTSW3,LIFTSW4,PLANEERR,PLANEERRSW,BETASW,
                 -
                 IYINITD,DRHOBIAS,VELBIAS1,VELBIAS2

COMMON(PLOTFL) T,QBAR,GLOAD1,ALT,GAMMA,GI,HA,HP,DRAG,DRAGDOT,
 QDOT,TEMP,HS,INCL1,LOD1,PHI1,PHIC1,ALTERR,VRELERR,RDOTERR,
 ICNT,ROLLERR,ROLLUNDER,KRHOWV,DRHO,VIEX,HS1,GPLLM,HSD,
 RDTERO,KRDT,RDTNM,KV,K1,K2,TEMPA   ,KHTOT,BOTOT,GWTOT
                                  21

COMMON(DISTURB)  NBOLGI, NKHELM, NATMO, NDRAG, SUMRHO,
                 -
                 RVACP

COMMON(PLOT2)  FILEMODE,NDATA,SUMPLTLOC,HIRESPLOC,
               FILEPLT, FILECNT,FILEFREQ,MCRLONUM,
               PRTLVL,TPHASE,TEND

INDEX I, J
```

```
/* READ IN THE INPUT PARAMETERS


DRIVER    SUBROUTINE
          READ WLIM,WDOTLIM,OMEGAD,INCL,INCLD,THETA,RY,VY
          READ CB, CBBIAS, LOD, LODBIAS, LODNAV, CBNAV

          READ DTSIM, PRTNO, TMAX, MCPLSW
          READ GAIN, FSW

          READ RNDDENS, RHOBIAS
          READ HA,HP,HANAV,HPNAV
          READ HEI,HEINAV,GUIDRATE,ICNTL,PLOTSW
          READ NAVSW,LODSW,RDTERO,KRDT
          READ BETASW
          READ SWITCH3BS,LIFTSW1,LIFTSW2,LIFTSW3,LIFTSW4
          READ PLANEERRSW,OPTION
          READ KRHOWV,VIEX,HS1,GPLLM
          READ RDTNM,KV,K1,K2
          READ STARTALT, SIZE
          READ THETANAV,DR,DH,DVR
          READ TSIZE, FILEMODE, ERRSW, PRTLVL, FILEFREQ
          READ MCRLONUM, MCRLOEND, MULTPERT, MULTERRR
          READ NATMO, NGRAVW, NKHELM, NBOLGI
          READ ALTCBIAS, ALTBLBIAS, ALTBMBIAS
          READ DRHOBIAS,VELBIAS1,VELBIAS2

          HPI = HPNAV
          RHOVAR = 1 + RHOBIAS
          PLANEERR = ABS(INCL - INCLD)

 /* PRINT THE INPUT PARAMETERS
          PRINT MSG, SP3
          ********** AEROBRAKING SIMULATOR **********
          PRINT MSG, SP2
          RUN CONTROL VARIABLES AND INITIAL INCLINATION
          PRINT HDG, MCRLONUM,OPTION,DRHOBIAS,VELBIAS1,VELBIAS2,SP4
          MCRLONUM   OPTION   DRHOBIAS  VELBIAS1  VELBIAS2
          PRINT FORMAT 501, MULTERRR, ERRSW,
                            PRTLVL, MCPLSW, FILEMODE,GAIN,FSW,
                          SWITCH3BS, MULTPERT, SP3
          LONG  FORMAT 501
          PRINT   MC      FILE    GAIN      FILTER   SWITCH3BS
          LEVEL  PLOTSW  MODE               FREQ       (DEG)
            $$      $       $      $.$$$      $$$       $$.$$
                                          1 -    PERT/ NO NAVERR
          MULTERRR $.$$     ERRSW: $    2 - NO PERT/    NAVERR
          MULTPERT $.$$                 3 -    PERT/    NAVERR
          PRINT FORMAT 502, LIFTSW1,LIFTSW2,LIFTSW3,LIFTSW4,PLANEERRSW,
                            BETASW
          FORMAT 502
          LIFTSW1     LIFTSW2  LIFTSW3  LIFTSW4   PLANEERRSW   BETASW
           $.$$$       $.$$$    $.$$$     $.$$$    $.$$$$        $$.$
```

```
PRINT HDG
WLIM      WDOTLIM   OMEGAD    INCLD     THETA     THETANAV
PRINT WLIM,WDOTLIM,OMEGAD,INCLD,THETA,THETANAV,SP2
PRINT HDG, CB ,CBBIAS, CBNAV, LOD, LODBIAS, LODNAV, SP2
W/CDA     CBBIAS    CBNAV     LOD       LODBIAS   LODNAV
PRINT HDG, DTSIM, PRTNO, TMAX, FILEMODE, FILEFREQ, SP2
DTSIM     PRTNO     TMAX      FILEMODE  FILEFREQ
PRINT HDG, RNDDENS, RHOVAR, RHOBIAS,SP2
RNDDENS   RHOVAR    RHOBIAS
PRINT HDG
HA        HP        HANAV     HPNAV     HP-HPNAV
PRINT HA,HP,HANAV,HPNAV,(HP-HPNAV),SP2
PRINT HDG
HEI       HEINAV    HEI-HEINAVGUIDRATE  ICNTL     PLOTSW
PRINT HEI,HEINAV,(HEI-HEINAV),GUIDRATE,ICNTL,PLOTSW,SP2
PRINT HDG
NAVSW     LODSW     RDTERO    KRDT
PRINT NAVSW,LODSW,RDTERO,KRDT,SP2
PRINT HDG
KRHOWV    VIEX      HS1       GPLLM
PRINT KRHOWV,VIEX,HS1,GPLLM,SP2
PRINT HDG
RDTNM     KV        K1        K2
PRINT RDTNM,KV,K1,K2,SP2
PRINT HDG
STARTALT  SIZE      DWNRNGERR HERROR    RADVELERR
PRINT STARTALT,SIZE,DR,DH,DVR,SP2
PRINT HDG, TSIZE, NATMO, NGRAVW, NBOLGI, NKHELM, SP2
TSIZE     NATMO     NGRAVW    NBOLGI    NKHELM
```

```
/* INITIALIZATION PROCESS

                        16
      MU = 1.40764685 10
      RE = 20925784
                    -6
      J2 = 1082.7 10
                        -5
      WE = 7.29211585 10
      GZERO = 32.146437
      DTR   = DEGTORAD
      FPNM  = 6076.115
      NMPF  = 1/FPNM

      GAMBIAS = DTR GAMBIAS
      -
      IPOLE = (0, 0, 1)
      DUM1 = 0, DUM2 = 0, DUM3 = 0, DUM4 = 0
      RNAV = HEINAV + RE
      R = HEI + RE
      SO = SIN(DEGTORAD OMEGAD)
      CO = COS(DEGTORAD OMEGAD)
      SID = SIN(DEGTORAD INCLD)
      SI = SIN(DEGTORAD INCL)
      CID = COS(DEGTORAD INCLD)
      CI = COS(DEGTORAD INCL)
      ST = SIN(DEGTORAD THETA)
      CT = COS(DEGTORAD THETA)
      STNAV = SIN(DEGTORAD THETANAV)
      CTNAV = COS(DEGTORAD THETANAV)
      -
      IYINITD= (SO SID, (-CO SID), CID)
      -      -
      IYD    = IYINITD
      INCLDB = INCLD
      NODEDB = OMEGAD

      IF OPTION = 2,
           SET FILE READ (10000),
                      -
           FILE READ MOMVEC,NODEDB,INCLDB,
           -          -
           IYD = UNIT(MOMVEC)
                 -
      PRINT HDG, IYD,INCLDB,NODEDB
      IYD                              INCLDB    NODEDB
                 -
      PRINT HDG, IYINITD
      IYINITD
      -
      IY = (SO SI, (-CO SI), CI)
      -
      RUNIT = (CT CO - ST CI SO, CT SO + CO CI ST, SI ST)
```

```
-
RUNITNAV = (CTNAV CO - STNAV CI SO, CTNAV SO + CO CI STNAV,
  SI STNAV)
-      -
RD = R RUNIT
-              -
RDNAV = RNAV RUNITNAV
-           -       -
R = R UNIT(RD + RY IY)
-                   -            -
RNAV = RNAV UNIT(RDNAV + RY IY)
RPNAV = HPNAV 6076.115 + RE
RANAV = HANAV 6076.115 + RE
RA = HA 6076.115 + RE
RP = HP 6076.115 + RE
ANAV = (RANAV + RPNAV) / 2
A = (RA + RP) / 2
VNAV = SQRT(MU(2/RNAV - 1/ANAV))
V = SQRT(MU(2/R - 1/A))
GNAV =-ARCCOS(SQRT(RANAV RPNAV/(RNAV(RANAV+RPNAV-RNAV))))
G =-ARCCOS(SQRT(RA RP/(R(RA+RP-R))))
-
VUNIT = ((-CO ST - SO CI CT), CO CI CT - SO ST, SI CT)
-
VUNITNAV = ((-CO STNAV - SO CI CTNAV), CO CI CTNAV - SO STNAV,
  SI CTNAV)
-      -
V = V VUNIT
-              -
VNAV = VNAV VUNITNAV
-           -       -
V = V UNIT(V + VY IY)
-                   -            -
VNAV = VNAV UNIT(VNAV + VY IY)
*           -         -         -    -
MG = (UNIT(R), UNIT(V), UNIT(V * R))
*                -            -              -       -
MGNAV = (UNIT(RNAV), UNIT(VNAV), UNIT(VNAV * RNAV))
-      * T
V = V MG   (SIN(G), COS(G), 0)
-            *     T
VNAV = VNAV MGNAV   (SIN(GNAV), COS(GNAV), 0)
IF DR = 0, IF DH = 0, IF DVR = 0, GO TO SAME
-
R = (RNAV   + DH 6076.115, RNAV   + DR 6076.115, 0)
          0                        1
-
V = (VNAV   + DVR, VNAV    , 0)
          0             1
                                 -    -     -
SAME     INC = RADTODEG ARCCOS(UNIT(R * V)  . IPOLE)
                                        -       -       -
         INCNAV = RADTODEG ARCCOS(UNIT(RNAV * VNAV)  . IPOLE)
```

```
PRINT HDG
RY          VY          INC         INCNAV
PRINT RY, VY, INC, INCNAV
PRINT SKIP

TEMP1 = NATMO
TEMP2 = NGRAVW
TEMP3 = NKHELM
TEMP4 = NBOLGI

IF MCRLONUM = 0,
   NGRAVW   = 0,
   NKHELM   = 0,
   NBOLGI   = 0,
   NATMO    = 0

RNAV = R, VNAV = V

CALL SWS.CONICS, 5, 0, MU, 0,(-1), R, V

RESUME FLAG1, TIMETOP,  RVACP, VVACP

PRINT FORMAT 100, (RVACP NMPF)
FORMAT 100
RVACP = ( $.$$$$$$$E$$    $.$$$$$$$E$$  $.$$$$$$$E$$) NM

CALL GCH.ORBITEL, MU, R, V
RESUME RVA, RVP, AV
PRINT FORMAT 101,((RVA - RE) NMPF),((RVP - RE) NMPF),
                 (AV NMPF), SP4
FORMAT 101
ALT VAC APOGEE  =  $.$$$$$E$$ NM
ALT VAC PERIGEE =  $.$$$$$E$$ NM
VAC SMA         =  $.$$$$$E$$ NM
```

```
/* RANDOM ERRORS SECTION


        DO TO CYCRND FOR I=1 (1) 10 ABS (MCRLONUM - 1)
CYCRND  DUM = RNDMN (1)

/* PERSERVE THE INITIAL NOMINAL STATE
        -           -           -           -
        RNOMINIT = R,           VNOMINIT = V
        -           -           -           -
        RNAVNOMI = R,           VNAVNOMI = V
        LODNOM   = LOD,         CBNOM    = CB

        IF RNDDENS = 0,
           RHOVAR = RHOVAR,
                  OTHERWISE IF MCRLONUM NOTEQ 0,
                            RHOVAR = RNDMN (RHOBIAS) + 1,
                            OTHERWISE RHOVAR = 1

        PRINT FORMAT 104, ((RHOVAR - 1) 100), SP3
              FORMAT 104
        THE LEVEL OF CONSTANT DENSITY BIAS FOR THIS RUN
        IS  $$$.$$$$ %

        IF LODNOM < 0,
           LODVAR = 1,
           OTHERWISE IF MCRLONUM NOTEQ 0,
                     LODVAR = 1 + RNDMN (LODBIAS),
                     OTHERWISE LODVAR =1
        LOD = LODVAR ABS (LODNOM)
        PRINT FORMAT 105, LOD, ((LODVAR - 1) 100),SP3
        FORMAT 105
        THE CONSTANT L/D FOR THIS RUN IS: $$$.$$$$$
        WHICH IS  $$$.$$$$$% FROM THE NOM. VALUE

        IF CBNOM < 0,
           CBVAR = 1,
           OTHERWISE IF MCRLONUM NOTEQ 0,
                     CBVAR = 1 + RNDMN (CBBIAS),
                     OTHERWISE CBVAR =1
        CB = CBVAR ABS (CBNOM)
        PRINT FORMAT 106, CB, ((CBVAR - 1) 100),SP3
        FORMAT 106
        THE CONSTANT W/CDA FOR THIS RUN IS: $$$.$$$$$
        WHICH IS  $$$.$$$$$% FROM THE NOM. VALUE

        IF FILEMODE = 2, SET FILE WRITE 80000

        IF MCRLONUM NOTEQ 0,
           NATMO  = TEMP1,
           NGRAVW = TEMP2,
           NKHELM = TEMP3,
           NBOLGI = TEMP4
```

```
SUMRHO    = 0, NDATA    = 0
NDRAG     = 0, FILEPLT = 0, FILECNT = 0
FIRSTPASS = 0, ISTART  = 0
SUMPLTLOC = 90000 + 600 MCRLONUM
HIRESPLOC = 20000

CALL GCH.SIMT7

IHDES = RNAV*VNAV
SET FILE WRITE(10000)
FILEWRITE IHDES
IF FILEMODE = 3,
SET FILE WRITE (90000 + 600 MCRLONUM - 1),
FILEWRITE FILEPLT
PRINT HDG, IHDES
IHDES:
AVGRHOBS = SUMRHO/NDRAG
FILECNT = FILECNT - 1

PRINT FORMAT 120, NDATA, AVGRHOBS, RHOVAR,FILECNT,
                  FILEPLT, SP4
LONG FORMAT 120
!!!   !!!   !!!   !!!   !!!   !!!   !!!   !!!   !!!
NDATA = $$$$$$$   AVGRHOBS = $.$$$$$  RHOVAR = $.$$$$$
!!!   !!!   !!!   !!!   !!!   !!!   !!!   !!!   !!!
!!!   !!!   !!!   !!!   !!!   !!!   !!!   !!!   !!!
FILECNT = $$$$    FILEPLT = $$$$
!!!   !!!   !!!   !!!   !!!   !!!   !!!   !!!   !!!

/* DO POST-AEROBRAKING BURNS

CALL GCH.BURNS2(BEGIN1),T,RNAV,VNAV,INCLDB,NODEDB
DO PRTSTARS

/* DO PLOTTING FOR INDIVIDUAL RUNS

IF MCPLSW >=1 AND FILEMODE = 3,
   CALL MCPLOT1
IF MCPLSW >= 1 AND FILEMODE = 4,
   CALL MCPLOT2
IF MCPLSW >= 1 AND FILEMODE = 5,
   CALL MCPLOT3
IF MCPLSW >= 1 AND FILEMODE = 7,
   CALL MCPLOT4
IF MCPLSW >= 1 AND FILEMODE = 8,
   CALL MCPLOT5
IF MCPLSW >= 1 AND FILEMODE >= 9,
   CALL MCPLOT6
MCRLONUM = MCRLONUM + 1
IF MCRLONUM <= MCRLOEND, GO TO MCRLO
IF MCPLSW >= 1,
```

```
      CALL FILEPLOT(ENDPLOT)

      RESUME
      RETURN


PRTSTARS PRINT FORMAT 900
      LONG  FORMAT 900

      ************************************************************************

      ************************************************************************


      START AT DRIVER
```

```
MAC* GCH.SIMT7
*****************************************************************
SOURCE      : GCH1752.THESIS.MAC(SIMT7)
AUTHOR      : H.R. MORTH AND G.C. HERMAN
PURPOSE     : SIMULATE AEROBRAKING FOR OTV
INPUTS      : INITIAL POSITION, VELOCITY, AND CONTROL VARIABLES
OUTPUTS     : STATE AND CONTROL VARIABLES DURING AEROBRAKING
*****************************************************************
                -
COMMON(CONST)  DUM1,IPOLE,DUM2,MU,RE,J2,DUM3,DUM4,WE,GZERO

COMMON(CABRAKE), GLOAD, WLIM, WDOTLIM, RNAV, VRELNAV, VNAV,
 INCL, INCLD, LODNAV, LOD, CB, DTSIM, PHIC, ROLL, RHO,
                                                  -  -
 RHOVAR, PLOTSWITCH, HSTEMP, KRHO, ICNTL, PRTNO, R, V,
 -     -
 RNAV, VNAV, HPI, TMAX, GUIDRATE, NAVSW, LODSW, RDOTNAV, VEX1,
 -
 IYD, STARTALT, SIZE, NGRAVW, TOUT, FIRSTPASS, ISTART,
 -
 ACCEL, CBNAV, RHOSTD, LODEST,SWITCH3BS,LIFTSW1,
                                                  -
 LIFTSW2,LIFTSW3,LIFTSW4,PLANEERR,PLANEERRSW,BETASW,IYINITD,
 DRHOBIAS,VELBIAS1,VELBIAS2

COMMON(PLOTFL) T,QBAR,GLOAD1,ALT,GAMMA,GI,HA,HP,DRAG,DRAGDOT,
 QDOT,TEMP,HS,INCL1,LOD1,PHI,PHIC1,ALTERR,VRELERR,RDOTERR,
 ICNT,ROLLERR,ROLLUNDER,KRHOWV,DRHO,VIEX,HS1,GPLLM,HSD,
 RDTERO,KRDT,RDTNM,KV,K1,K2,V,VREL,VIDES,GINAV,GREL,GRELNAV,
 GIDES,HANAV,HAD,HPNAV,HPD,RY,THETAR,VY,THETAV,DELTA,ANGTONODE,
 ANGTOAPOGE,HAPRECISE,DELTAVCIRC,DELTAVPLAN,ALTNAV,
 KHTOT,BOTOT,GWTOT

COMMON(PRINT)  VAR   ,LAT,LONG,VELENG,CONTROLMODE,
                  13
               BETANAV,S2ROLL,SWITCH2,INCDOT,NODEDOT,
               ALPHADOT,IRATE1,NODERATE1,
               ALPHARATE1,X1,LIFTM,WDGDES,THETARNAV,
               IERROR,NODEERR,ALPHAERR,ALPHAERRMF,THETAVDES,
               ALPHANAV,TRUEANNAV

COMMON(DISTURB) NBOLGI,NKHELM, NATMO, NDRAG, SUMRHO,
                 -
                 RVACP

COMMON(PLOT2)  FILEMODE,NDATA,SUMPLTLOC,HIRESPLOC,
               FILEPLT, FILECNT,FILEFREQ,MCRLONUM,
               PRTLVL,TPHASE,TEND

COMMON(COMP)   RDOTDO, DRGRF,DERROR,GAMMAREF,DV1,VRELNAV1,
               DRGNOM,DRGM,CD,KDRAG,GAIN,CDDOT,FSW

INDEX I,J,N
DIMENSION (DT,4),(SWDOT,4)
```

```
/* SIMULATOR INITIALIZATION


AEROSIM  SUBROUTINE
         FTPNM = 6076.115
         NMPFT = 1/FTPNM
         IF FIRSTPASS = 1, GO TO SIMLOOP
         FIRSTPASS = 1, DT=DTSIM ,PSW = 0, T = 0, TOUT = T,
                        PHIC =  90, PHASE = 1,T = 0,
                        PHI  = PHIC, GUIDCOUNT = GUIDRATE,
                        DTSAVE = DT, DO SETUP, ROLLUNDER = 0,
                                      4
                        C0=-4.79519468 10   , C1=0.99700549,
                                       -6                    -12
                        C2=-4.17893612 10   , C3=5.39401157 10    ,
                                                            -5
                        H0 = 207040, RHO0 = 1.3096315 10    ,
            DPHI/DT=0, RHOOLD = 1, ALTOLD = 1,  IP = 0,
            GIDES = 0.74532027803482129336747559l0,
            SPHI = SIN(DEGTORAD PHI),
            CPHI = COS(DEGTORAD PHI),
            HAD = 150.0, HPD = 40.864252277879033958666591O
            IF NATMO = 0, PRINT SKIP
            IF NATMO = 0, GO TO SIMLOOP
            CALL REP.USOTV62(INIT)

          PRINT SKIP
```

```
 /* START OF SIMULATION


SIMLOOP  DO AERO
                    -
         ACCEL = ABVAL(ACCEL)
                              2
         QBAR = .5 RHO VREL   / GZERO
         -    -
         IA = ACCEL / ACCEL
         GLOAD = ACCEL / GZERO
         GLOAD1 = GLOAD
         LOD1 = LOD
                  -
NOLOD    R = ABVAL(R)
                   -
         RNAV = ABVAL(RNAV)
         ALT = R - RE
         ALTNAV = RNAV - RE
                  -      -
         RDOTNAV = VNAV . RNAV / RNAV
                -   -
         RDOT = V . R / R
                                 -   -
         GAMMA = RADTODEG ARCSIN(R . VREL / (VREL R))
                  -
         V = ABVAL(V)
                   -                                    -
         VNAV = ABVAL(VNAV)
         -        -          -        -
         VRELNAV = VNAV - WE( IPOLE * RNAV)
                          -
         VRELNAV = ABVAL(VRELNAV)
                   -   -
         GI = ARCSIN(R . V / (V R))
                        -      -
         GINAV = ARCSIN(RNAV . VNAV / (VNAV RNAV))
                                    -      -       -
         AINCL = RADTODEG ARCCOS(UNIT(RNAV * VNAV) . IPOLE)
                                  2      2
         X = SQRT(1-(R V COS(GI))  (2/R-V /MU)/MU)
                                                2              2
         XNAV = SQRT(1-(RNAV VNAV COS(GINAV))  (2/RNAV-VNAV /MU)/MU)
                                       2
         HA =(R(1 + X)  MU / (2 MU - R V ) - RE)/6076.115
                                                 2
         HANAV =(RNAV(1+XNAV)MU/(2 MU - RNAV VNAV ) - RE)/6076.115
                                       2
         HP =(R(1 - X)  MU / (2 MU - R V ) - RE)/6076.115
                                                 2
         HPNAV =(RNAV(1-XNAV)MU/(2 MU - RNAV VNAV ) - RE)/6076.115
         GI = GI RADTODEG
         GINAV = GINAV RADTODEG
```

```
          GAMMANAV = RADTODEG ARCSIN(RNAV . VRELNAV/(VRELNAV RNAV))
          INCL = AINCL
          INCL1 = INCL
                    -           -
          RDOTNAV = VNAV . UNIT(RNAV)
NOPRT2    IF PLOTSWITCH NZ, DO SAVE
          ALTERR = ALTNAV - ALT
          VRELERR = VRELNAV - VREL
          RDOTERR = RDOTNAV - RDOT

/* CALL TO GUIDANCE
          IF GUIDCOUNT = GUIDRATE, GUIDCOUNT = 0,
             CALL GCH.GUID8C, RESUME
          GUIDCOUNT = GUIDCOUNT + 1
          IF PSW = 0, DO PRNTDTA
          PSW = PSW + 1
          IF PSW = PRTNO, PSW = 0
          ROLLERR = PHIC - PHI
          PHIC1 = PHIC
          W=DPHI/DT

 /* CALL TO CONTROL
          CALL RAYS.AUTOP PHIC, PHI, W, ROLLUNDER
                    -  -
          RESUME DT,SWDOT
          DO TO SLOOP1 FOR N = 0(1)3
          SWDOT = SWDOT
                        N
          DT = DT
                 N
          IF DT  = 0, IF DT  = 0, IF DT  = 0, DPHI/DT = 0
               0            1            2
          IF DT <= 0, IF N = 3, GO TO LOOP2
          IF DT <= 0, GO TO SLOOP1
```

```
          2      2
INTEG     D PHI/DT  = SWDOT WDOTLIM
INTEG1    DO TO LOOP1 FOR I = 0(1)3
          ROLLERR = PHIC - PHI
          IF ABS(ROLLERR) > 180, ROLLERR = ROLLERR - 360 SIGN(ROLLERR)
          SERR = SIGN(ROLLERR)
          IF FIRSTPASS = 0, FIRSTPASS = 1
ROLLCMD   SPHI = SIN(DEGTORAD PHI)
          CPHI = COS(DEGTORAD PHI)
AERO      DO TO AEROEND
          -      -      -       -
          VREL = V - WE (IPOLE * R)
          -          -          -       -
          VRELNAV = VNAV - WE (IPOLE * RNAV)
                     -
          VREL = ABVAL(VREL)
                           -
          VRELNAV = ABVAL(VRELNAV)
          -          -
          IX = UNIT(VREL)
          -          -    -
          IZ = UNIT (IX * R)
          -          -    -           -
          IY = UNIT(IZ * IX) CPHI + IZ SPHI
                                      -      -
          CALL JPH.USATM62, 0, (.3048 R), WE, IPOLE
          RESUME RHO
                            3
RHOCALC   RHO = RHO (.3048 ) / 0.45359237
          RHOSTD = RHO
          IF NATMO = 0, RHOFAC2 = 1.
DRG       DRHO = RHOVAR RHOFAC2
          IF VNAV < VIEX + VELBIAS2,DRHO = DRHOBIAS
          IF VNAV < VIEX + VELBIAS1,DRHO = RHOVAR RHOFAC2
          RHO = DRHO RHO


                           2
          DRAG = .5 RHO VREL  / CB
          LIFT = LOD DRAG
          -                -          -
AEROEND   ACCEL = -DRAG IX + LIFT IY
          -           -          -  3
          GRAV = -MU R/ (ABVAL(R))
           -      -
          DR/DT = V
           -      -      -
          DV/DT = GRAV + ACCEL
          -           -           -     3
          GNAV = -MU RNAV/ (ABVAL(RNAV))
           -          -
          DRNAV/DT = VNAV
           -          -      -
          DVNAV/DT = GNAV + ACCEL
```

```
                                  -      -      -      -          2      2
          DIFEQ T, DT, DR/DT, DV/DT, DRNAV/DT, DVNAV/DT, D PHI/DT
          W = DPHI/DT
LOOP1     TNAV = T, TOUT = T
SLOOP1    TNAV = T
LOOP2     IF ALT > 400000, IF RDOT > 0, DO PLOTS, DO PRNTDTA, EXIT
          IF T > TMAX, DO PLOTS, DO PRNTDTA, EXIT

                                -
          DELTARHO = ABVAL(ACCEL)  (RHO - RHOSTD)/RHOSTD
          SUMRHO   = SUMRHO + DELTARHO
                                          -
          NDRAG    = NDRAG + ABVAL(ACCEL)

          IF ABS(PHI) > 180,
             PHI = PHI - SIGN(PHI) 360

          GO TO SIMLOOP

          RETURN
PLOTS     DO TO NDPLOTS
          IF FILEMODE >= 2,
             GO TO NDPLOTS
          SET FILE WRITE 902
          FILE WRITE ICNT
NDPLOTS   RESUME
SAVE      RESUME
SETUP     RESUME
```

```
/* CALCULATE PRINT PARAMETERS

PRNTDTA  DO TO NDPRNT
                                                                    3.15
         QDOT = 17600 SQRT(RHO / (.0027 GZERO)) (VREL / 26000)
                                          -10 .25
         TEMP = (778.158 QDOT / 3.74 10    )      - 460
         ALTOLD = ALT, RHOOLD = RHO
         - -   - -
         IYA = UNIT(R * V)
               - -
         RY = IYD . R
              - -
         VY = IYD . V
              -   -
         LY = IYD . (LIFT IY)
              -     -
         DY = IYD . (-DRAG IX)
              - -
         GY = IYD . GRAV
                    2       2
         VHT = SQRT(V  - RDOT )
         THETAR = RADTODEG RY/R
         THETAV = RADTODEG VY/VHT
                                 - -
         DELTA = RADTODEG ARCCOS(IYA . IYD)
                                   - -
         PLANECHNG = RADTODEG ARCCOS(IYA . IYINITD)
                   - -      -9
         IF ABVAL(IYA * IYD) < 10   , ANGTONODE = 0, GO TO CON,
                   -  -     -  -     -
           OTHERWISE NODE = (IYA * IYD)/ABVAL(IYA * IYD)
                     -          -
         IF UNIT(R)  = UNIT(NODE), ANGTONODE = 0, GO TO CON
                     -          -
         IF UNIT(R)  = -UNIT(NODE), ANGTONODE = 180, GO TO CON
         - -   - -
         IN = UNIT(R * NODE)
            -      -  -   -
         IF IN NOTEQ IYA, NODE = -NODE
                   -      -
         IF ABS(NODE . R/R) > 1, ANGTONODE = RADTODEG
                        -      -
           ARCCOS(SIGN(NODE . R/R)), GO TO CON
                                  - -
         ANGTONODE = RADTODEG ARCCOS(NODE . R/R)

                        -
CON      RVACP = ABVAL(RVACP)

                        -
         R     = ABVAL(R)
```

```
          ALTEST = ABVAL(R) - RE

          AIP = RADTODEG ARCCOS(RVACP . R/(RVACP R))

          IF RVACP . R < 0 , AIP = - AIP

          CALL SWS.CONICS, 5, 0, MU, 0, 1, R, V

          RESUME FLAG, TIMETOAPOGEE, RA, VA

          RA = ABVAL(RA)

          ANGTOAPOGE = RADTODEG ARCCOS(R . RA/(RA R))
          HAPRECISE = (RA - RE)/6076.115
          IF GI NEG, ANGTOAPOGE = 360 - ANGTOAPOGE

          VA = ABVAL(VA)
          AT = (RA + RE + 150 6076.115)/2
          VTATRA = SQRT(MU(2/RA - 1/AT))
          BURN1 = VTATRA - VA
          VCIRC = SQRT(MU/(RE + 150 6076.115))
          VTATRP = SQRT(MU(2/(RE + 150 6076.115) - 1/AT))
          BURN2 = ABS(VCIRC - VTATRP)
          DELTAVCIRC = BURN1 + BURN2

          INC = RADTODEG ARCCOS(UNIT(R * V) . IPOLE)
          DELTAVPLAN = VCIRC TAN(DEGTORAD DELTA)

          IV = UNIT(V)

          IR = UNIT(R)
          IF (IV  IR  - IR  IV ) = 0, OMEGA = 0, GO TO NEXT
              0   2     0   2
          OMEGA = RADTODEG ARCTAN((IV  IR -IR  IV )/(IV  IR -IR  IV ))
                                   1   2   1   2     0   2   0   2


          IF PRTLVL = 1, DO PRT1
          IF PRTLVL = 8, DO PRT8
          IF PRTLVL =12, DO PRT12
          IF PRTLVL =13, DO PRT13

NDPRNT    RESUME
```

```
/* PRINT ROUTINES


/* PRINT LEVEL 1

PRT1     DO TO NPRT1
NEXT     IF IPRT >= 2,
           IPRT = 0,
           PRINT SKIP
         IPRT = IPRT + 1
         PRINT HDG
         T          RHO/RHOSTD  BOTOT    GWTOT      KHTOT
         PRINT T, (RHO/RHOSTD), BOTOT, GWTOT, KHTOT,SP1
         PRINT HDG
         VI         VINAV      VREL      VRELNAV    VIDES-VI
         PRINT V, VNAV, VREL, VRELNAV, (VEX1 - V), SP1
         PRINT HDG
         GI         GINAV      GREL      GRELNAV    GIDES-GI  AIP
         PRINT GI, GINAV, GAMMA, GAMMANAV, (GIDES - GI), AIP, SP1
         PRINT HDG
         HA         HANAV      HAD-HA    HP         HPNAV      HPD-HP
         PRINT HA, HANAV, (HAD - HA), HP, HPNAV, (HPD - HP), SP1
         PRINT HDG
         RY         THETAR     VY        THETAV     DELTA      OMEGA
         PRINT RY, THETAR, VY, THETAV, DELTA, OMEGA, SP1
         PRINT HDG
         ANGTONODE ANGTOAPOGEHAPRECISE DELTAVCIRCDELTAVPLANINC
         PRINT ANGTONODE, ANGTOAPOGE, HAPRECISE, DELTAVCIRC,
               DELTAVPLAN, INC, SP1
         PRINT HDG
         ALT        RDOT       HS        QDOT       TEMP
         PRINT ALT, RDOT, HS,  QDOT, TEMP, SP1
         PRINT HDG
         RHO        DRAG       GLOAD     ROLL       ROLLC      ROLLRATE
         PRINT RHO , DRAG, GLOAD, PHI, PHIC, W, SP1
         PRINT HDG
         INCLNAV    INCLD      INCN-INCLDLOD         LODEST     DRHO
         PRINT AINCL, INCLD, (AINCL-INCLD), LOD, LODEST, DRHO, SP1
         PRINT HDG
         LY         DY         GY        RDOTNAV    RDOT-RDTNVALTNAV
         PRINT LY, DY, GY,RDOTNAV,(RDOT-RDOTNAV),ALTNAV, SP1
         PRINT HDG
         X          Y          ANGLAT    YG         YU         YL
         PRINT VAR  TO VAR
                  0        5
         PRINT HDG
         GYNAV      TGO        ANGERR    ANGERRP
         PRINT VAR  TO VAR
                  6        9
         PRINT HDG
         RAT        DVEX       VEX1      RAT33      RDTDRV     RDOTERR
         PRINT VAR   , VAR   , VEX1, VAR   , VAR   , RDOTERR, SP3
                  10      11            12      13
NPRT1    RESUME
```

```
 /* PRINT LEVEL 8
PRT8     DO TO NPRT8
         IP = IP + 1
         IF IP =1,
             PRINT FORMAT 1037
         LONG FORMAT 1037
         T(S)|ALT(FT)| ROLLC    ROLL    RATE|  DRGEST  DRGREF    DERROR
         | RDOTNAV    RDOTREF    RDOTERR| HA(NM)      HP(NM) |  DRHO

         PRINT FORMAT 1036,T, ALT, PHIC, PHI, W, CD, DRGRF,DERROR,
                           RDOTNAV, RDOTDO,RDOTERR, HA, HP,
                           DRHO
         LONG  FORMAT 1036
         $$$$| $$$$$$| $$$.$  $$$.$  $$.$$| $$$.$$$  $$$.$$$  $$$.$$$
         | $$$$.$$   $$$$.$$   $$$$.$$| $$$$$.$$$  $$$$.$$|  $.$$$$
         IF 10 TRUNCATE(ABS(IP)/10) = ABS(IP),
            PRINT BLANK
         IF 50 TRUNCATE(ABS(IP)/50) = ABS(IP),
            PRINT BLANK, SKIP,
            PRINT FORMAT 1037
 NPRT8    RESUME

 /* PRINT LEVEL 12
PRT12    DO TO NPRT12
         IP = IP+1
         RTD = RADTODEG
                                - -
         CALL GCH.ORBITS3 MU,R,V
         RESUME RA,RP,OMEGA,ARGLAT,DUMMY,LONGNODE,SEMIA,ECC,
                                    - - -
                ANGMOM,INCL,EN,ARGW,IE,IN,IH,DUMMY,WEDGE
         HA1= (RA -RE)NMPFT
         HP1= (RP -RE)NMPFT
         SEMIA = SEMIA NMPFT
         ROLL  = PHI DEGTORAD
         IF OMEGA > PI, OMEGA = OMEGA -2 PI
         IF LONGNODE > PI, LONGNODE = LONGNODE -2 PI
                        -
         R      = ABVAL(R)
         WEDGE    = RTD WEDGE
         INCL     = RTD INCL
         OMEGA    = RTD OMEGA
         LONGNODE= RTD LONGNODE
         THETAVDES = RTD THETAVDES
         VAR        = RTD VAR
            1               1
         VAR        = RTD VAR
            2               2
         VAR        = RTD VAR
            3               3
         VAR        = RTD VAR
            4               4
```

```
          VAR       = RTD VAR
             5              5
          VAR       = RTD VAR
             8              8
          PRINT HDG,T,ALT,HA1,HP1,DELTA,(RTD WDGDES)
          TIME(S)   ALT(FT)   HA(NM)    HP(NM)    DELTA(D)  WDGDES(D)
          PRINT HDG, PHIC,PHI,VY,THETAV,THETARNAV,RY
          PHIC(D)   PHI(D)    VY(FT/S)  THETAV(D) THETAR(D) RY

          PRINT HDG, VAR ,VAR ,VAR ,VAR ,VAR ,ALPHADOT
                        1   3   4   5   8
          Y(D)      YG(D)     YU(D)     YL(D)     ANGERR(D) ALPHADT(D)
          PRINT HDG, INCL,LONGNODE,TRUEANNAV,THETAVDES,IRATE1,ALPHARATE1
          INCLD     LNGNODED  TRUEANNAVDTHVDESD   IRATED    ALPRATED
          PRINT HDG, OMEGA,ALPHANAV,BETANAV,VNAV,LIFT,DRAG
          OMEGAD    ALPHANAVD BETANAVD  VNAV      LIFT      DRAG
          PRINT HDG, CONTROLMODE,VAR ,S2ROLL,SWITCH2,X1,PLANECHNG
                                    0
          CMODE     RVFLAG    S2ROLL    SWITCH2   X1        PLANEC(D)
          PRINT HDG,LIFTM,DRGM,ACCEL
          LIFTM     DRGM      ACCEL
          PRINT HDG, RDOT,RDOTDO,RDOTERR,DRHO,W,SP3
          RDOT      RDOTDO    RDOTERR   DRHO      ROLLRATE

          IF 3 TRUNCATE(ABS(IP)/3) = ABS(IP),
             PRINT BLANK, SKIP

 /* PRINT LEVEL 13
 NPRT12   RESUME
PRT13     DO TO NPRT13
          IP = IP+1
                                - -
          CALL GCH.ORBITS3 MU,R,V
          RESUME RA,RP,OMEGA,ARGLAT,TRUEAN,LONGNODE,SEMIA,ECC,
                                      - - -
                 ANGMOM,INCL,EN,ARGW,IE,IN,IH,ALPHA,WEDGE,BETA
          HA1= (RA -RE)NMPFT
          HP1= (RP -RE)NMPFT
          SEMIA = SEMIA NMPFT
          ROLL  = PHI DEGTORAD
          IF OMEGA > PI, OMEGA = OMEGA -2 PI
          IF LONGNODE > PI, LONGNODE = LONGNODE -2 PI
          IF TRUEAN > PI, TRUEAN = TRUEAN -2 PI
                      -
          R      = ABVAL(R)
          ALPHA    = RADTODEG ALPHA
          WEDGE    = RADTODEG WEDGE
          BETA     = RADTODEG BETA
          INCL     = RADTODEG INCL
          OMEGA    = RADTODEG OMEGA
          TRUEAN   = RADTODEG TRUEAN
          LONGNODE= RADTODEG LONGNODE
          ANGLATD = RADTODEG VAR
                               2
```

```
          PRINT HDG,T,ALT,HA1,HP1,WEDGE,(RADTODEG WDGDES)
          TIME(S)    ALT(FT)    HA(NM)     HP(NM)     WEDGE(D)   WDGDES(D)
          PRINT HDG, PHIC,PHI,ALPHAERRMF,RDOT,RDOTDO,RY
          PHIC(D)    PHI(D)     ALPERRMF  RDOT       RDOTREF    RY
          PRINT HDG,ANGLATD,ALPHAERR,THETARNAV,NODEDOT,INCDOT,ALPHADOT
          ANGLAT(D)  ALPHAERR   THETARNAV NDDOT(D)   IDOT(D)    ALPDOT(D)
          PRINT HDG, INCL,LONGNODE,TRUEAN,NODERATE1,IRATE1,ALPHARATE1
          INCL(D)    LNGNODE(D)TRUEAN(D)  NDRATE(D)  IRATE(D)   ALRATE(D)
          PRINT HDG, OMEGA,ALPHA,BETA,VNAV,LIFT,DRAG
          OMEGA(D)   ALPHA(D)   BETA(D)    VNAV       LIFT       DRAG
          PRINT HDG, CONTROLMODE,VAR ,S2ROLL,SWITCH2,X1,ANGMOM,SP3
                                    0
          CMODE      RVFLAG     S2ROLL     SWITCH2    X1         ANGMOM

          IF  4 TRUNCATE(ABS(IP)/4)  = ABS(IP),
             PRINT BLANK, SKIP
 NPRT13    RESUME

PRTSTARS  PRINT FORMAT 900
          LONG  FORMAT 900

          **********************************************************************

          **********************************************************************

PRTDASHS  PRINT FORMAT 901
          LONG  FORMAT 901

          ----------------------------------------------------------------

          ----------------------------------------------------------------

          START AT AEROSIM
```

```
MAC*GCH.GUID8C
*************************************************************************
          SOURCE    : GCH1752.THESIS.MAC(GUID8C)
          AUTHOR    : H.R. MORTH  AND G.C. HERMAN
          PURPOSE   : GUIDANCE LAW FOR AEROBRAKING LIFT-MODULATED OTV
                      AND LATERAL GUIDANCE ALGORITHM
          INPUTS    : ALTITUDE, ALTITUDE RATE, VELOCITY, AND THE
                      ACCELEROMETER MEASUREMENTS
          OUTPUTS   : MAGNITUDE AND SIGN OF THE COMMANDED ROLL ANGLE
          COMMENTS  : AEROBRAKING GUIDANCE LAW IS ESSENTIALLY THE ONE
                      DESCRIBED IN REFERENCE [3 AND 4]
                      A LATERAL GUIDANCE ALGORITHM BASED ON CONTOLLING
                      THE HINGE LINE POSITION IS IMPLEMENTED(SEE
                      LATCTL)
*************************************************************************

          COMMON(CABRAKE),
           CGLOAD,TEMP1,TEMP2,RNAV,DUMV,VNAV,AINCLD,INCLTG,LODNAV,TEMP3,
           TEMP4,DTSIM,ROLLCD,ROLDEG,FANS3,TEMPA  ,RNAV,VNAV,TEMB  ,GRATE,
                                                11                1
           TEMP6,LODSW,RDOTNAV,VEX1,IYD,STARTALT,SIZE,NGRAVW,
           TOUT, FIRSTPASS, ISTART, ACCEL, CBNAV, RHOSTD, LODEST,
           SWITCH3BS,LIFTSW1,LIFTSW2,LIFTSW3,LIFTSW4,PLANEERR,
           PLANEERRSW,BETASW

          COMMON(CONST),
           DUM1,IPOLE,DUM2,MU,RE,J2,DUM3,DUM4,WE,GS,GS1,ALT1

          COMMON(PLOTFL),
           T,QBAR,GLOAD,ALTT,GAMMA,GI,HA,HP,DRAG,DRAGDOT,
           QDOT,TEMP,HS,INCL,LOD,PHI,PHIC,ALTERR,VRELERR,RDOTERR,
           ICNT,ROLLERR,ROLLUNDER,KRHOWV,DRHO,VEX,HS1,GPLLM      ,HSD,
           RDTERO,KRDT,RDTNM,KV,K1,K2,VIT,VREL,VIDES,GINAV,GREL,GRELNAV,
           GIDES,HANAV,HAD,HPNAV,HPD,RY,THETAR,VY,THETAV,DELTA,ANGTONODE,
           ANGTOAPOGE,HAPRECISE,DELTAVCIRC,DELTAVPLAN,ALT,KHTOT,
           BOTOT,GWTOT

          COMMON(PRINT),
           X,Y,ANGLAT,YG,YU,YL,GY,TGO,ANGERR,ANGERRP,LODC,DVEX,LODRDTE,
           RDOTDRV, LAT, LONG, VELENG,CONTROLMODE,BETA,S2ROLL,SWITCH2,
           INCDOT,NODEDOT,ALPHADOT,IRATED,NODERATED,
           ALPHARATED,X1,LIFTM,WDGDES,THETARNAV,IERROR,NODEERR,
           ALPHAERR,ALPHAERRMF,THETAVDES,ALPHA,TRUEAN

          COMMON(PLOT2)  FILEMODE,NDATA,SUMPLTLOC,HIRESPLOC,
                         FILEPLT, FILECNT,FILEFREQ,MCRLONUM,
                         PRTLVL,TPHASE,TEND

          COMMON(COMP)   RDOTRF, DRGRF,DERROR,GAMMAREF,DV1,VRELNAV,
                         DRGNOM,DRGM,DRGEST,KDRAG,GAIN,DRGDOT,FSW
```

```
ABRAKE   SUBROUTINE

 /* COMPUTE NAVIGATED THETAV
                 -     -
         VYNAV = IYD . VNAV
                       2         2
         VH = SQRT(VNAV - RDOTNAV )
         THETAVNAV = RADTODEG VYNAV/VH
         IF ISTART = 0,
            DO ICS
         IF VNAV < VQUIT, EXIT
         DO GPARAMS
         IF (ACCELM - ACCELSTRT) > 0.0,
            IGUIDE = 1
         IF IGUIDE = 1, IF IEXIT = 0,
            DO EGCTL
         IF VNAV < (VEX + VIFNL),
            IEXIT = 1
         IF IEXIT = 1,
            DO UPCTL
         DO LATCTL
 /* PLOT COUNTER
         GPLCT = GPLCT + 1
         IF GPLCT = GPLLM,
            DO MAKEFL
ENDBRAKE RETURN ROLLC
```

```
/* INPUT GUIDANCE PARAMETERS BELOW

ICS       DO TO NDICS
            RTD = RADTODEG
            TSTEP = DTSIM GRATE
            IGUIDE = 0, IEXIT = 0, KFLAG1 = 0, IFILE = 0, ISTART = 1
            VSAT = 25766.1973, VQUIT = 25000.
            HS = 20650
            DRGRFBS = 22.6, ACCELSTRT = .05 GS
            DRGRFMIN = 0.10 GS, DMAX = 4.0 GS
            BSQ = 2000 2000, DAMP1 = .75, OMEGA = PI/50
            VIO = VNAV
            YBIAS = 0.0008725, GNLAT = 1.5
            ROLL = 15/RTD
            S2ROLL = SIGN(THETAVNAV), X = 0, ILAT = 0
            ANGERRP = 0
            VIFNL = 5500.0
            LODEST = LODNAV, DRGNOMOLD = 0
            VSO1 = (GNLAT LODNAV)/ RTD
            RATCMN = LODNAV COS(15/RTD)
            LODC = 0
            RDTMAX = 2000, RDTMIN = 150
            GS1 = 31, ALTF = 400000, S20 = SIN(20/RTD)
            GPLCT = GPLLM - 1
 /* NO LATERAL CONTROL IF ICNTL = 0
           ICNTL = TEMPA
                       4
            DTR = DEGTORAD
            RTD = RADTODEG
            DRGRFDOT=0,  RDOTRF = 0
            DRGRF2 = 0 , KDRAG = 1, FPASS = 0,PHASE = 1
NDICS       DVEX = 0, VIMIN = 27000
```

```
 /* GENERATE GUIDANCE PARAMETERS


GPARAMS  DO TO NDGPAR

 /* RADIUS MAGNITUDE
         R1 = RE + ALT
 /* NORMALIZED VELOCITY-SQ
         VSO = (VNAV VNAV) / (VSAT VSAT)
 /* LIFT FOR EQUILIBRIUM
         LFTEQ = (VSO - 1.0) GS

         ACCELM = ABVAL(ACCEL)

         VRELNAV = VNAV - WE(IPOLE*RNAV), VRELNAV = ABVAL(VRELNAV)

         DRGM = ABS(UNIT(VRELNAV) . ACCEL)

         IF CGLOAD < LODSW OR LODSW = 0, GO TO NOLOD
                                2       2
         LIFTM = SQRT(ACCELM -  DRGM )
         LODM  = LIFTM/DRGM
         LODEST = .9 LODEST + .1 LODM

 /* FILTER FOR DRAG
                                      2
NOLOD    DRGNOM = .5 RHOSTD VRELNAV /CBNAV

         IF ALT > 320000, FREQ = 2,
            OTHERWISE FREQ = FSW
         IF FPASS = FREQ, FPASS = 0,
         KDRAG  = (1 - GAIN)KDRAG  + GAIN DRGM/( DRGNOM)
         FPASS = FPASS + 1
         DRGEST = KDRAG DRGNOM

 /* FIND THE MEASURED DRAG RATE
         DRGDOT = KDRAG (DRGNOM - DRGNOMOLD) / TSTEP
         DRGNOMOLD = DRGNOM
         AA1 = DRGDOT / DRGEST
         BB1 = 2.0 (DRGEST / VRELNAV)
         RDOTDRV = -HS (AA1 + BB1)
         DRGEQ = -LFTEQ / (-LODEST - (RDOTNAV / VRELNAV))
         IF DRGEQ < 0, DRGEQ = 0
         DRGRF = 0, DERROR = 0, C16 = 0, C17 = 0, DAMP=0
NDGPAR   LODRF = 0, LODDRGE = 0, LODRDTE = 0
```

```
 /* EQUILIBRIUM GLIDE CONTROL


EGCTL     DO TO NDEGCTL
          DRGRF = DRGEQ + DRGRFBS
          IF KFLAG1 = 0, DRGRF2 = DRGRF, KFLAG1 = 1
          DRGRFDOT = (DRGRF - DRGRF2) / TSTEP, DRGRF2 = DRGRF
          RDOTRF = -HS (DRGRFDOT / DRGRF + 2 (DRGRF / VRELNAV))
                                                          2
          DAMP = DAMP1 SQRT(1.0 + ((RDOTNAV - RDOTRF) / BSQ))
          IF DRGRF < DRGRFMIN, DRGRF = DRGRFMIN
          IF DRGRF > DMAX, DRGRF = DMAX
          AK1 = (HS / (DRGRF DRGRF)) (OMEGA OMEGA
                                        2
                - 3.0 (DRGRFDOT / DRGRF)
                + 3.0 (DRGRFDOT / VNAV)
                                    2
                - 4.0 (DRGRF / VNAV)
                + (LFTEQ / HS))
          AK2 = (HS / (DRGRF DRGRF)) (2.0 DAMP OMEGA
                - 3.0 (DRGRF / VNAV)
                + 2.0 (DRGRFDOT / DRGRF))
          C17 = AK2 DRGRF / HS
          C16 = AK1 + AK2 ((DRGRFDOT / DRGRF) - 2.0 (DRGRF / VNAV))
          LODRF = (-LFTEQ / DRGRF)
                  + (HS / DRGRF) ((DRGRFDOT / DRGRF) (DRGRFDOT / DRGRF)
                  - 3.0 (DRGRFDOT / VNAV)
                  - 4.0 (DRGRF / VNAV) (DRGRF / VNAV))
          DERROR = DRGEST - DRGRF
          LODDRGE = C16 DERROR
          RDOTERR = RDOTNAV - RDOTRF
          LODRDTE = -C17 RDOTERR
          TPHASE = T
 /* COMMANDED VERTICAL L/D
NDEGCTL   LODC = LODRF + LODDRGE + LODRDTE
```

```
 /* UPCONTROL PHASE


UPCTL     DO TO NDUPCTL

          DVPE = GS1  (ALTF - ALT) /VNAV
          VEX1 = VEX -DVEX
          DV1 = VNAV - VEX1 - DVPE
          IF DV1 = 0, DV1 = 1

 /* REFERENCE RDOT FOR UPCONTROL
          RDOTRF = DRGEST (VEX1/VNAV) HS1/DV1
          IF RDOTRF > RDTMAX, RDOTRF = RDTMAX
          IF RDOTRF < RDTMIN, RDOTRF = RDTMIN
          IF VNAV < VEX1 , RDOTRF = RDTMAX
 /* CORRECTION TO DESIRED EXIT VEL.
          IF VNAV > VIMIN,
             DVEX = KV (RDOTRF - RDTNM)
          IF DRGEST < DRGRFMIN,
                       C17 = K2/DRGRFMIN,
             OTHERWISE C17 = K2/DRGEST

 /* MORE PRECISE LIFT FOR EQUILIBRIUM
          LFTEQ1 = (VNAV VNAV - MU/R1)/R1
          IF DRGEST < DRGRFMIN,
                       LODRF = - LFTEQ1/DRGRFMIN,
             OTHERWISE LODRF = - LFTEQ1/DRGEST
          RDOTERR = RDOTNAV - RDOTRF
          IF ABS(RDOTERR) < 15, C17 = K1 C17
 /* L/D FOR RDOT ERROR
          LODRDTE = -C17 RDOTERR
          PHASE=0
 /* COMMANDED VERTICAL L/D
NDUPCTL   LODC = LODRF + LODRDTE
```

```
 /* LATERAL CONTROL LOGIC


LATCTL   DO TO NDLAT
         TEND   = T
         TGOMAX = 500
         -        -    -
         IH = UNIT(RNAV*VNAV)
                     -  -
         WEDGE = ARCCOS(IH.IYD)
                          -    -    -
         CALL GCH.ORBITS4A MU,RNAV,VNAV,IYD
         RESUME BETA,ALPHA,INC,W,TRUEAN, ANGMOM,NODE,ARGLAT
         PHID = PHI DTR
         IF NODE > PI, NODE = NODE -2 PI
         IF TRUEAN > PI, TRUEAN = TRUEAN -2 PI

 /* COMPUTE THE RATE OF CHANGE OF THE LONGITUDE OF THE HINGE LINE
         SFB  = SIN(TRUEAN - BETA)
         SW   = SIN(WEDGE)
         AD   = LIFTM SIN(PHID)
         IF SW = 0,ALPHARATE = 0,OTHERWISE
                   ALPHARATE   = RNAV SFB AD/(ANGMOM SW)
         ALPHARATED  = RTD ALPHARATE

         BETA     = RTD BETA
         ALPHA    = RTD ALPHA
         INC      = RTD INC
         W        = RTD W
         WEDGE    = RTD WEDGE
         TRUEAN   = RTD TRUEAN
         NODE     = RTD NODE

         IF ABS(LODC) < RATCMN,  LODC1 = LODC,
                       OTHERWISE LODC1 = RATCMN SIGN(LODC)
         IF RDOTNAV < 0, GO TO GCALC
         TGO = (ALTF - ALT)/RDOTNAV
         IF TGO > TGOMAX, TGO = TGOMAX

 /* GRAVITATION COMPENSATION SECTION
         -                -          -     3
GCALC    GRAVNAV = -MU RNAV/(ABVAL(RNAV))
              -     -
         GY = IYD . GRAVNAV
         YG = GY TGO / VH
         ANGLAT = DTR THETAVNAV
```

```
 /* PHASE PLANE BOX LIMITS SECTION
         Y = VSO1 + YBIAS
                 -     -
         RYNAV = IYD . RNAV
         THETARNAV = RTD RYNAV/RNAV

/* CALCULATE THE DESIRED VELOCITY OUT-OF-PLANE ERROR
         IF TRUEAN < 40,SIGMA = TRUEAN + 20,OTHERWISE SIGMA=TRUEAN
         IF VNAV < VEX + 1600.,Y=YBIAS,
             WDGDES = ARCSIN(SIN(DTR THETARNAV)/SIN(DTR SIGMA)),
             WDGDESD= RTD ABS(WDGDES),
             THETAVDES = PI/2 - ARCSIN(COS(WDGDES)/COS(DTR THETARNAV)),
             THETAVDES  =  SIGN(VYNAV) ABS(THETAVDES),
             THETAVDESD =  RTD THETAVDES

         YU = (Y - YG)
         YL = -Y - YG
         YM = -YG

 /*  REDEFINE ALPHA TO AVOID ENTERING EXCLUSION ZONE
         IF BETA > TRUEAN,ALPHA = ALPHA - 180

 /* ANGULAR ERROR
         ANGERR = ANGLAT - YM
         ANGERR1= ANGLAT - THETAVDES
         IF PLANEERR < PLANEERRSW,VEXBS=800,OTHERWISE VEXBS = 0
         IF VNAV < VEX + VEXBS, DO BETAC,OTHERWISE DO PLANEC

 /* ROLL ANGLE SECTION
RCALC    FADLD1 = LODC1 / LODNAV
         IF ABS(FADLD1) >= 1.0, FADLD1 = SIGN(LODC)
         ROLLC = S2ROLL ARCCOS(FADLD1)
         ROLLCD = RTD ROLLC
         S2ROLLOLD = S2ROLL
         ALPHAOLD= ALPHA
         BETAOLD = BETA
         INCOLD  = INC
         NODEOLD = NODE

 /* SAVE PREVIOUS ERROR
         ANGERRP1= ANGERR1
NDLAT    ANGERRP = ANGERR
```

```
 /* PLANE ERROR CONTROL


 PLANEC  DO TO NPLANEC
         CONTROLMODE = 1
         IF VNAV < VEX + 1600., CONTROLMODE =2

 /* REVERSAL FLAG OFF
         IF ANGLAT > YU OR ANGLAT < YL, IF ABS(ANGERR) <= ABS(ANGERRP),
           IF (ANGERR ANGERRP) > 0, X = 1
         IF ABS(LODC) < RATCMN, GO TO RREV
 /* HIGH IN-PLANE LIFT SECTION
         IF Y NOTEQ YBIAS, Y = Y / 2.0, YU = Y - YG, YL = -Y - YG,
           IF ANGLAT > YU OR ANGLAT < YL,
           IF ABS(ANGERR) <= ABS(ANGERRP),
           IF (ANGERR ANGERRP) > 0, X = 1

 /* REVERSAL FLAG ON
RREV     IF ANGLAT < YU AND ANGLAT > YL, ILAT = 1,
           IF ABS(ANGERR) <= ABS(ANGERRP), X = 0

 /* ROLL REVERSAL CHECK
         IF ANGLAT >= YU OR ANGLAT <= YL, IF X = 0, IF ICNTL NOTEQ 0,
           IF ILAT NOTEQ 0, S2ROLL = -S2ROLL, X = 1
NPLANEC  RESUME

 /* MODIFIED PLANE ERROR CONTROL LIMITS
PLANEC2  DO TO NPLNEC2
         Y=YBIAS
         YU =  Y + THETAVDES
         YL = -Y + THETAVDES
         YM =  THETAVDES
         ANGERR = ANGERR1
         ANGERRP = ANGERRP1
NPLNEC2  RESUME
```

```
/* HINGE LINE CONTROL


BETAC    DO TO NBETAC
         DO PLANEC2
         SWITCH2 = BETA BETAOLD
         SWITCH3 = (BETA - TRUEAN)
         X1           = 1
         IF SWITCH3 > 0,IF ALPHA < ALPHAOLD,
                      X1=0
         IF SWITCH3 < 0,IF ALPHA > ALPHAOLD,
                      X1=0
         IF ABS(LIFTM SIN(PHID)) < LIFTSW2 ,DO BETAC2,
                              OTHERWISE DO BETAC1
NBETAC   RESUME

BETAC1   DO TO NBETAC1

 /* ETA CONTROL PHASE
         MF           = 1
         IF WDGDESD < .01,DO PLANEC,GO TO NBETAC
         IF SWITCH3 < 0,IF BETA > BETASW,IF X1 = 0,
                IF ABS(ALPHARATED) < .3,CONTROLMODE=8,
                S2ROLL = -S2ROLLOLD,X1=1,X=1,GO TO NBETAC
         IF SWITCH3 > 0,IF ABS(ALPHARATED) < LIFTSW1,
                              MF = 8 6/SWITCH3BS
         IF SWITCH3 < 0,IF ABS(ALPHARATED) < LIFTSW1,
                              MF = 4 6/SWITCH3BS
BETAC1A  DO TO NBETAC1A

         IF CONTROLMODE NOTEQ 3, X = 1
         CONTROLMODE = 3

         IF ABS(SWITCH3) < MF SWITCH3BS,IF X1=0,
                      S2ROLL = -S2ROLLOLD,X1=1,X=1

NBETAC1A RESUME
         IF WEDGE > 1.5 WDGDESD,IF X1 = 0,X=0,DO PLANEC
NBETAC1  RESUME
```

```
/* BETA CONTROL PHASE


BETAC2   DO TO NBETAC2
         IF WDGDESD < .001,IF ABS(ALPHARATED) > .2,MF=1,DO PLANEC,
                        GO TO NBETAC
         IF WDGDESD < .01,IF ABS(ALPHARATED) > 1.5,MF=1,DO BETAC1A,
                        GO TO NBETAC
         IF CONTROLMODE < 4,X=0
         IF ALPHA < W, IF ALPHA > ALPHAOLD,X=1
         IF ALPHA > W, IF ALPHA < ALPHAOLD,X=1
         UBETA= BETASW,LBETA=-20,CONTROLMODE=4
         IF ABS(ALPHARATED) < LIFTSW3 ,UBETA=BETASW/2,
                          LBETA = -10,CONTROLMODE=5
         IF ABS(ALPHARATED) < LIFTSW4 ,UBETA= 1,LBETA=- 2,
                                       CONTROLMODE=6
         IF TRUEAN > BETA,IF SWITCH = 0,X=0,SWITCH=1
         IF SWITCH2 < 0,IF BETAOLD < TRUEAN,X=0
         IF BETA > UBETA OR  BETA < LBETA,IF X=0,
                              S2ROLL= -S2ROLLOLD,X=1
         IF BETA > TRUEAN,IF ABS(BETA) < ABS(BETAOLD),
                              S2ROLL= -S2ROLLOLD,X=1
NBETAC2  RESUME
```

```
 /* FILE DATA


MAKEFL   DO TO NDMKFL
         NDATA = NDATA + 1
         FILEPASS = 1 + FILEPASS
         PHI1 = PHI
         PHIC1 = PHIC
         IF ABS(PHI1) > 180, PHI1 = PHI1 - SIGN(PHI1) 360
         IF ABS(PHIC1) > 180, PHIC1 = PHIC1 - SIGN(PHIC1) 360
         IF FILEMODE = 10,
            FILECNT = FILECNT + 1,
            SET FILEWRITE HIRESPLOC,
            FILEWRITE T,DRHO, ALT, PHI1, PHIC1, BETA, W,
                      ALPHA, WEDGE, INC,THETAV,THETAR,
                      IERROR,NODEERR,ALPHAERRMF,WDGDESD,
                      THETAVDESD,(RTD YU),(RTD YL),
            HIRESPLOC = HIRESPLOC + 19,TPHASE=0,
            GO TO NSKIP1
         IF (FILEMODE = 9 AND PHASE = 1),GO TO NSKIP1
         IF FILEMODE = 9,
            FILECNT = FILECNT + 1,
            SET FILEWRITE HIRESPLOC,
            FILEWRITE T,DRHO, ALT, PHI1, PHIC1, BETA, W,
                      ALPHA, WEDGE, INC,THETAV,THETAR,
                      IERROR,NODEERR,ALPHAERRMF,WDGDESD,
                      THETAVDESD,(RTD YU),(RTD YL),
            HIRESPLOC = HIRESPLOC + 19
NSKIP1   ICNT=ICNT+1
         GPLCT = 0
NDMKFL   RESUME
FILEI    DO TO NDFILEI
         IF FILEMODE >= 2, GO TO NDFILEI
         SET FILE WRITE(900)
         ICNT = 0
         ILOC = 2000, NVARS = 72, NCYC = 2000
         FILE WRITE ILOC,NVARS,NCYC
         SET FILE WRITE ILOC
NDFILEI  IFILE = 1

         START AT ABRAKE
```

```
MAC* GCH.ORBITS4A
*****************************************************************
SOURCE       : GCH1752.THESIS.MAC(ORBITS4A)
AUTHOR       : G.C. HERMAN
PURPOSE      : COMPUTES THE ORBITAL ELEMENTS AND THE CONTROL
               PARAMETERS
INPUTS       : CURRENT RADIUS AND VELOCITY VECTORS AND THE
               DIRECTION OF THE ANGULAR MOMENTUM VECTOR OF
               THE DESIRED ORBIT
OUTPUTS      : THE ORBITAL ELEMENTS COMPUTED ARE:
                    SEMI-MAJOR AXIS       A
                    ECCENTRICITY          E
                    INCLINATION           I
                    LONGITUDE OF THE
                     ASCENDING NODE       LONGNODE
                    AGRUMENT OF PERIAPSIS W
             IN ADDITION THE FOLLOWING ORBITAL PARAMETERS ARE FOUND:

                    TRUE ANOMALY          F
                    AGRUMENT OF LATTITUDE THETA
                    ANGULAR MOMENTUM      H
                    SEMI-LECTUS RECTUM    P
                    RADIUS OF PERIGEE     RP
                    RADIUS OF APOGEE      RA
                    TOTAL ENERGY          EN
               THE CONTROL PARAMETERS COMPUTED ARE:
                    ALPHA
                    BETA
*****************************************************************
                          - - -
ORBIT     SUBROUTINE MU,R,V,IHNOM
          -                   -
          IPOLE = (0,0,1) , IX =(1,0,0)
                  -               -
          R=ABVAL(R), V = ABVAL(V)
          -         -   -         -
          IR = UNIT(R), IV = UNIT(V)
          -    -   -              -    -         -
          H  = R * V  ,  H = ABVAL(H) , IH = UNIT(H)
          P = H H/MU
          AINV = (2/R) - (V V/MU)
                               6
          IF AINV = 0,  A = 10 ,
                        OTHERWISE A = 1/AINV
                      -       -
          I = ARCCOS(IPOLE . IH)

 /* UNIT VECTOR ALONG NODES
          -     -     -
          IN = (IPOLE*IH)/SIN(I)
          -     - -     -
          E  = (V*H -MU IR)/MU
                     -    -         -
          E  = ABVAL(E),  IE = UNIT(E)
```

```
        EN = -MU/(2 A)
        T  = 2 PI A SQRT(A/MU)
        RA = A(1 + E)
        RP = A(1 - E)
                                - -
        LONGNODE = ARCCOS(IX.IN)
        IF IN < 0, LONGNODE = 2 PI - LONGNODE
             1
                                - -
        W        = ARCCOS(IE.IN)
        IF IE  < 0, W = - W
             2
                  - -
        ARGW     = IE.IN
                                - -
        F        = ARCCOS(IR.IE)
           - -
        IF IR.IV < 0, F = 2 PI - F
        THETA    = W + F
/* FIND PLANE ERROR PARAMETERS
        -       -     -
        INTER = IHNOM*IH
                            -     -          -              -
        ALPHA = ARCCOS((INTER.IN)/(ABVAL(INTER)  ABVAL(IN)))
                              -       - -
        IF ALPHA > PI/2, INTER = IH*IHNOM,
                            -     -          -              -
        ALPHA = ARCCOS((INTER.IN)/(ABVAL(INTER)  ABVAL(IN)))
        IF INTER  < 0, ALPHA = - ALPHA
                2
/* DEFINE ANGLE BETWEEN APSIDAL LINE AND HINGE LINE
        BETA = ALPHA - W
        RETURN BETA, ALPHA,I,W,F,H,LONGNODE,THETA
        START AT ORBIT
```

List Of References

1. Morth, H.R., "A Guidance Law for Aerobraking a Ballute Type Vehicle", CSDL Memo hrm 84-3, Cambridge, MA June 6, 1984.

2. Steinberg, M.J., "An Analytic Predictor/Corrector Aerobraking Algorithm for a Drag-Modulating Ballute-Type Orbital Transfer Vehicle", MIT Thesis, Cambridge, MA September, 1985.

3. Morth, H.R., "An Analytic Aerobraking Guidance Law for a Low L/D Vehicle", CSDL Memo 10E-84-02, Cambridge, MA June 26, 1984.

4. Hill, O., "An Adaptive Guidance Logic for an Aeroassisted Orbital Transfer Vehicle", AAS paper 83-357, 1983.

5. Cruz, M.I., "Trajectory Optimization and Closed Loop Guidance of Aeroassisted Orbital Transfer", AAS paper 83-415, 1983.

6. Steinberg, M.J., "Plane Control Logic of an Analytic Aerobraking guidance Law for a Low L/D Vehicle", CSDL Memo 10E-84-13, November 5, 1984.

7. Battin, R.H., "An Introduction to the Mathematics and Methods of Astrodynamics", in press, May 1986.

8. Burning, H., Lecture Notes for Astrodynamics, University of Michigan, Ann Arbor, MI, September, 1983.

9. Hill, O., "Proposed Atmosphereic Guidance Logic for a Maneuvering Orbital Transfer Vehicle", Memo JSC-18620, 82-FM-58, NASA Mission Planning and Analysis Division, Lyndon B. Johnson Space Center, Houston, TX, October, 1982.

10. Robertson, W.M., "Values of Gravity Wave, Kelvin-Helmhotz Instability, and Bolgiano Layer Parameters", CSDL OTV Memo 10E-85-02, Cambridge, MA January, 1985.

11. Higgins, J.P., " An Aerobraking Guidance Concept for a Low L/D AOTV", CSDL OTV Memo 10E-84-04, Cambridge, MA May, 1984.